

**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA
SOUZA**

ETEC DA ZONA LESTE

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

**Kevin dos Santos Farias
Laura Nogueira Santos
Rafael de Alcantara França
Raissa Amato Gonzaga**

GETJOB: Plataforma de Vídeos Online

**SÃO PAULO
2022**

**Kevin dos Santos Farias
Laura Nogueira Santos
Rafael de Alcantara França
Raissa Amato Gonzaga**

GETJOB: Plataforma de Vídeos Online

Trabalho de Conclusão de Curso apresentado ao Curso do Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas - AMS, orientado pelo Professor Wagner de Oliveira Lucca, como requisito parcial para obtenção do título de Técnico em Desenvolvimento de Sistemas.

**SÃO PAULO
2022**

AGRADECIMENTOS

A

Profa. Vilma Cardoso dos Santos

Pelo incentivo e apoio.

Coord. Rogério Bezerra Costa

Pela paciência e cooperatividade.

Prof. Wagner de Oliveira Lucca

Pelo suporte no desenvolvimento do projeto.

Prof. Ediney Ciasi Barreto

Pelo suporte no desenvolvimento do referencial teórico.

Prof. Marlon Marques da Silva

Pelo incentivo e apoio psicológico.

Luca Domingues Rossetto

Pelo apoio e incentivo.

Daniel Ideriba

Pelo suporte no desenvolvimento da programação.

RESUMO

Através de uma plataforma online dedicada a compartilhar conhecimentos sobre o mercado de trabalho, visamos contribuir com a inserção de mais jovens no mesmo. Mediante estudo em *Unified Modeling Language* (UML), *Hypertext Markup Language* (HTML), *Cascading Style Sheet* (CSS), *JavaScript* e Banco de Dados, a plataforma realizara o cadastro de usuários e fornecera vídeos personalizados. Também permitirá a publicação de vídeos de terceiros, contando com a relação do conteúdo com o projeto.

Palavras-Chave: Desemprego. Educação. Emprego. Jovens. Trabalho.

ABSTRACT

Through an online platform dedicated to sharing knowledge about the job market, we aim to contribute to the insertion of more young people in it. Through studies in Unified Modeling Language (UML), Hypertext Markup Language (HTML), Cascading Style Sheet (CSS), JavaScript and Database, the platform will register users and provide personalized videos. It will also allow the publication of videos by third parties, relying on the relationship of the content with the project.

Keywords: *Unemployment. Education. Employment. Younger. Work.*

LISTA DE ILUSTRAÇÕES

Figura 1 - Cadastro do Usuário, Código HTML	9
Figura 2 - Cadastro do Usuário, Página Web.....	10
Figura 3 - Código CSS.....	12
Figura 4 - Cadastro do Usuário Estilizado, Página Web	12
Figura 5 - Código JavaScript	14
Figura 6 - Cadastro do Usuário Funcional, Página Web	14
Figura 7 - Cadastro do Usuário Funcional, Página Web (2).....	15
Figura 8 - Cadastro do Usuário Funcional, Página Web (3).....	15
Figura 9 - Organização de Dados.....	16
Figura 10 - Tabela Cliente ÑN.....	18
Figura 11 - Tabela Cliente ÑN (2)	18
Figura 12 - Tabela Cliente e Telefone Normalizadas na 1FN.....	19
Figura 13 - Tabela Venda Não Normalizada na 2FN.....	19
Figura 14 - Tabela Venda e Produto Normalizadas na 2FN.....	20
Figura 15 - Tabela Venda Não Normalizada na 3FN.....	20
Figura 16 - Tabela Venda Normalizada na 3FN	20
Figura 17 - Símbolos do Dicionário de Dados.....	21
Figura 18 - Exemplo de Dicionário de Dados	21
Figura 19 - Simulação de um diagrama ER.....	22
Figura 20 - Opções de Serviços.....	23
Figura 21 - Organização em Pastas	24
Figura 22 - Organização de Pastas no Storage	24
Figura 23 - Hospedar Projeto no Firebase.....	25
Figura 24 – Exemplo de Sistema de Controle Bancário	26
Figura 25 – Relacionamento Entre os Atores.....	27
Figura 26 – Comunicação Entre Atores e Casos de Uso.....	28
Figura 27 - Relacionamento de Inclusão	29
Figura 28 – Exemplo de Relacionamento de Extensão	29
Figura 29 - Exemplo de Generalização	30
Figura 30 - Pontos de Extensão	30
Figura 31 - Exemplo de Documentação de Caso de Uso.....	31
Figura 32 - Símbolos de início e fim	32
Figura 33 - Execução das atividades.....	32
Figura 34 - Decisões Durante o Fluxo.....	32
Figura 35 - Paralelismo no Diagrama de Atividades.....	33
Figura 36 - Fluxo com Objetos.....	33
Figura 37 - Diagrama de Atividade da Criação de um Artigo	34
Figura 38 - Carro Funcionando.....	34
Figura 39 - Diagrama de Casos de Uso.....	35
Figura 40 - Diagrama de Atividade: Efetuar Cadastro.....	36
Figura 41 - Diagrama de Atividade: Efetuar Login.....	37
Figura 42 - Diagrama de Atividade: Cadastro de Conta do Produtor.....	38

Figura 43 - Diagrama de Atividade: Editar Perfil.....	39
Figura 44 - Diagrama de Atividade: Publicar Vídeo.....	40
Figura 45 - Diagrama de Atividade: Atividades do Vídeo.....	41
Figura 46 - Diagrama de Estado	42
Figura 47 - DER	42
Figura 48 – Tela Inicial: Home	43
Figura 49 – Tela de Cadastro	44
Figura 50 – Tela de Login.....	44
Figura 51 – Principal Tela de Vídeos.....	45
Figura 52 - Tela de Busca.....	45
Figura 53 - Tela de Perfil.....	46
Figura 54 – Tela de Edição de Perfil	46
Figura 55 - Tela de Cadastro do Produtor	47
Figura 56 – Tela de Envio de Vídeo	47
Figura 57 - Tela de Vídeo.....	48

LISTA DE ABREVIATURAS E SIGLAS

Banco de Dados (BD)

Cascading Style Sheet (CSS)

Diagramas de Entidade Relacionamento (DER)

Dicionário de Dados (DD)

Geographic Information System (GIS)

HyperText Markup Language (HTML)

JavaScript (JS)

Massachusetts Institute of Technology (MIT)

Não Normalizada (ÑN)

Not Only Standard Query Language (NoSQL)

Object Modeling Technique (OMT)

Object-Oriented Software Engineerin (OOSE)

Online Analytical Processing (OLAP)

Organização Internacional do Trabalho (OIT)

Primeira Forma Normal (1FN)

Rational Software Process (RUP)

Segunda Forma Normal (2FN)

Sistema de Gerenciamento de Banco de Dados (SGBD)

Software Development Kit (SDK)

Standard Query Language (SQL)

Terceira Forma Normal (3FN)

Unified Modeling Language (UML)

SUMÁRIO

1	INTRODUÇÃO	7
2	REFERENCIAL TEÓRICO.....	8
2.1	HTML	8
2.2	CSS.....	10
2.3	JavaScript	12
2.4	Banco de Dados.....	15
2.4.1	Banco de Dados não relacional	16
2.4.2	Normalização	17
2.4.3	Dicionário de Dados.....	21
2.4.4	Diagramas ER.....	21
2.5	Firebase	22
2.5.1	Funções do Firebase	23
2.6	UML.....	25
2.6.1	Diagrama de Casos de Uso	26
2.6.2	Diagrama de Atividades	31
2.6.3	Diagrama de Máquina de Estado.....	34
3	DESENVOLVIMENTO	35
3.1	Diagrama de Casos de Uso	35
3.2	Diagrama de Atividades	35
3.3	Diagrama de Estado	41
3.4	DER	42
3.5	Telas	43
4	CONCLUSÃO	49
	REFERÊNCIAS	50

1 INTRODUÇÃO

Nos últimos tempos, ficou evidente como o desemprego afetou o país. Segundo a Organização Internacional do Trabalho (2021), na América Latina, a taxa média de desemprego de jovens entre 15 e 24 anos teria atingido 23,8% no primeiro trimestre de 2021. Dessa forma, os jovens têm se caracterizado como o contingente populacional mais atingido por esse fenômeno. Pela falta de acesso a um conteúdo que ensine o que o atual mercado de trabalho procura, esse problema apenas tende a crescer conforme o passar dos anos.

Observando como o futuro desses jovens será negativamente atingido pela falta de conhecimento profissional, ou noção de como o mercado de trabalho funciona, é notória a importância de ensinamentos que podem ser cruciais para o avanço em processos seletivos.

Através de pesquisas bibliográficas, estudo de *Unified Modeling Language* (UML), *HyperText Markup Language* (HTML), *Cascading Style Sheet* (CSS), *JavaScript* e Banco de Dados, foi desenvolvida uma plataforma de vídeos com o objetivo de compartilhar conteúdos que facilitem o ingresso dos jovens no mercado de trabalho.

As tecnologias designadas mostram-se capazes de suprir os objetivos impostos pela pesquisa, assim como no auxílio para a produção da plataforma, que se propõe uma plataforma de vídeos online, que permitirá, através do cadastro, acesso à conteúdos referentes ao desenvolvimento pessoal de cada usuários.

2 REFERENCIAL TEÓRICO

Esta seção apresenta o conteúdo teórico sobre as tecnologias envolvidas no desenvolvimento do projeto de pesquisa da plataforma de vídeos online GetJob.

2.1 HTML

Como explicado por Silva (2015), *Hypertext Markup Language* (HTML), Linguagem de Marcação de Hipertexto, permite que se defina qual maneira a informação será mostrada para o usuário. Hipertexto é todo texto inserido em um documento para a web e que tem como principal característica a possibilidade de se interligar a outros documentos da web com uso de links. As linguagens de marcação são utilizadas na web em setores que demandem a capacidade de comunicação entre dispositivos, sistemas e plataformas distintas.

Foi criada por Tim Berners-Lee, físico britânico, cientista da computação e professor do *Massachusetts Institute of Technology* (MIT), estimulada pelo sucesso da conexão entre um cliente e um servidor na internet, em dezembro de 1990. A ideia da linguagem era para compartilhar documentos científicos entre os acadêmicos, de acordo com o objetivo inicial da própria internet.

Segundo Flatschart (2011), HTML é a principal linguagem utilizada na web. Ela permite a criação de documentos estruturados em títulos, parágrafos, listas, links, tabelas, formulários e em muitos outros elementos nos quais podem ser incorporadas imagens e objetos como, por exemplo, uma animação ou um vídeo.

A importância da linguagem de marcação HTML é que ela define o que é cada informação. Sozinho, o sistema não entende visualmente o que é cada elemento. Conforme Bernardo (2022), o HTML trabalha juntamente com os navegadores da internet e os sistemas de mecanismos de busca. Depois que o código de uma página está pronto, o navegador interpreta as informações e renderiza o site visualmente de acordo com as instruções. Assim, a pessoa que acessa consegue visualizar as informações, disponíveis em forma de texto, formas, imagens e outros recursos.

Como disse Silva (2008), a representação da linguagem HTML é escrita nas *tags* delimitadas, que identificam a função e o conteúdo de cada elemento. A seguir vamos apresentar algumas *tags*:

- DOCTYPE: É uma declaração para informar qual é a versão do HTML utilizada no arquivo.
- html: Utiliza-se para definir a estrutura para aplicações web.
- body: Sua finalidade é definir o corpo do documento, ou seja, onde é escrito todas as suas funções, tudo que estiver dentro da *tag* body será mostrado de alguma forma no conteúdo da página.
- h1: É utilizada para destacar títulos e subtítulos de uma página. Utilizada na estrutura de uma página, facilita a leitura e entendimento.
- label: É utilizado para adicionar um rótulo ao formulário. Os rótulos servem para informar para que serve cada campo do formulário HTML. Representa uma legenda para um item em uma interface de usuário.
- input: É usado para criar controles interativos para formulários, receber dados do usuário.
- name: Deve ser usado para identificar campos de formulários. Sempre que enviamos um formulário, o atributo “name” é quem identifica o valor no servidor.
- br: Produz uma quebra de linha em um texto, ou seja, é o que vai separar as linhas na aplicação.
- button: É o que introduz botões as páginas.

Na Figura 1, é apresentado o código para a realização de uma página da internet com HTML, criada para a demonstração das *tags* mencionadas. Na Figura 2, o resultado do código:

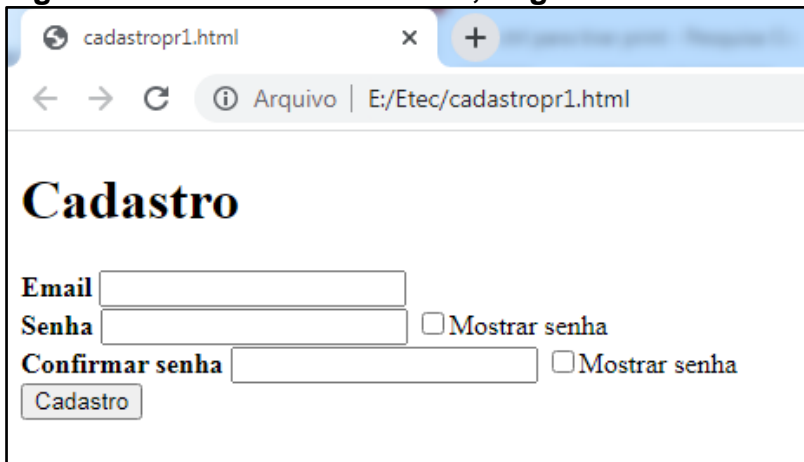
Figura 1 - Cadastro do Usuário, Código HTML

```

1  <!DOCTYPE html>
2  <html>
3  <body>
4  <h1>Cadastro</h1>
5
6  <label for="name"><b>Email</b></label>
7  <input type="text" id="email" name="email" />
8  <br />
9
10 <label for="name"><b>Senha</b></label>
11 <input type="password" id="senha" name="senha" />
12 <input type="checkbox" onclick="esconder1()" />Mostrar senha
13 <br />
14
15 <label for="name"><b>Confirmar senha</b></label>
16 <input type="password" id="senha2" name="senha2" />
17 <input type="checkbox" onclick="esconder2()" />Mostrar senha
18 <br />
19
20 <button onclick="cadastro()">Cadastrar</button>

```

Fonte: Autoria própria, 2022.

Figura 2 - Cadastro do Usuário, Página Web

The image shows a web browser window with a single tab titled 'cadastrop1.html'. The address bar displays 'Arquivo | E:/Etec/cadastrop1.html'. The main content area features a registration form with the following elements:

- Cadastro** (Registration) - Main heading
- Email** - Input field
- Senha** - Input field with a **Mostrar senha** (Show password) checkbox
- Confirmar senha** - Input field with a **Mostrar senha** (Show password) checkbox
- Cadastro** - Submit button

Fonte: Autoria própria, 2022.

2.2 CSS

Segundo Scheidt (2015), Cascading Style Sheet (CSS), Folha de Estilo em Cascata é uma linguagem voltada para a estilização em páginas web. O termo folha de estilo significa a descrição de um conjunto de regras que permitem definir a aparência de um website. Quando falamos em aparência, estamos nos referindo às diversas características existentes em uma página.

De acordo com Miletto (2014), enquanto o HTML passa a ser utilizado somente como elemento para estruturar as páginas, o CSS muda a forma de organização das páginas. Com o CSS, é possível definir em um único local a formatação que será utilizada por cada *tag*. Com isso, apenas um arquivo é alterado, sendo que a mudança é automaticamente propagada a todas as páginas que compõem o site.

Para que a aparência de sua página fique no estilo desejado, é preciso que suas linhas de código estejam dentro das regras e efetuando suas devidas funções. As regras no CSS possuem escopo global. Alterar uma regra pode gerar erros, portanto, é comum que novas regras sejam criadas e as regras antigas acabem ficando obsoletas porque, muitas vezes, evita-se editá-las ou removê-las. (SILVA, 2008).

Para exemplificar, alguns dos principais atributos e *tags* do CSS e suas funções:

- **margin**: Define a margem nos quatro lados de certo elemento.
- **padding**: Define a distância entre o conteúdo de um elemento e suas bordas.
- **border**: Serve para estilizar o site da melhor forma. Permite separar um espaço do layout de uma página, definindo onde vai aparecer.

- `font-family`: Permite que se faça uma lista de famílias de fontes e/ou nomes genéricos de famílias a serem especificados para um elemento selecionado.
- `height`: As propriedades `height`, `min-height` e `max-height` são usadas para definir a altura dos boxes.
- `colors`: Define o valor da cor de um elemento com seu conteúdo em texto e seu estilo.
- `var`: É usada para inserir o valor de uma propriedade personalizada substituindo qualquer parte do valor de outra propriedade.
- `background`: É um atalho para definir os valores de fundo individuais em um único lugar na folha de estilo.
- `display`: Especifica o tipo de caixa de renderização usada por um elemento.
- `divider`: É uma linha neutra que separa conteúdo.
- `layout`: Permite informar aos elementos onde devem ficar, qual o tamanho, quanto de espaçamento vai existir e criar lacunas.
- `container`: É um componente que garante que o seu layout vai ficar alinhado corretamente na página. Pode definir as margens laterais da página, ou deixar sem margens e, também, posiciona o conteúdo no centro do browser.
- `grid`: É um sistema de estruturação de layout. Permite configurar layouts em duas dimensões (linhas e colunas). A junção de linhas e colunas formam uma grade.
- `section`: É utilizada para marcar as seções de conteúdo de uma página, assim o agrupamento fica de forma lógica no conteúdo, separando a informação em áreas diferentes.

Na Figura 3, os atributos e *tags* citados anteriormente sendo utilizados para estilizar a mesma página de cadastro da Figura 2. E na Figura 4, o resultado da estilização:

Figura 3 - Código CSS

```

1  {
2  margin: 0;
3  padding: 0;
4  border: 0;
5  box-sizing: border-box;
6  font-family: "Poppins", sans-serif;
7  }
8
9  :root {
10 --header-height: 4.5rem;
11
12 /* colors */
13 --hue: 159;
14 /* HSL color mode */
15 --base-color: rgb(0, 110, 255);
16 --base-color-second: rgb(0, 110, 255);
17 --base-color-alt: hsl(var(--hue) 10% 53%);
18 --title-color: hsl(var(--hue) 41% 10%);
19 --text-color: hsl(0 0% 46%);
20 --text-color-light: hsl(0 0% 98%);
21 --body-color: hsl(0 0% 98%);
22
23 body {
24 font: 400 1rem 'DM Sans', sans-serif;
25 color: var(--text-color);
26 background: var(--body-color);
27 -webkit-font-smoothing: antialiased;
28 }
29
30 h1 {
31 font: 700 1.075rem 'Poppins', sans-serif;
32 color: var(--title-color);
33 -webkit-font-smoothing: auto;
34 }
35
36 .btn-link {
37 background-color: var(--base-color);
38 color: var(--text-color-light);
39 height: 3.5rem;
40 display: inline-flex;
41 align-items: center;
42 padding: 0 2rem;
43 border-radius: 0.25rem;
44 font: 500 1rem 'DM Sans', sans-serif;
45 transition: background 0.3s;
46
47
48 .divider-1 {
49 height: 1px;
50 background: linear-gradient(
51 270deg,
52 hsla(var(--hue), 36%, 57%, 1) 0%,
53 hsla(var(--hue), 65%, 88%, 0.34) 100%
54 );
55 }
56
57 /*==== LAYOUT ===== */
58 .container {
59 margin-left: 1.5rem;
60 margin-right: 1.5rem;
61 }
62
63 .grid {
64 margin-left: 1.5rem;
65
66 display: grid;
67 gap: 2rem;
68 }
69
70 .section {
71 padding: 5rem 0;
72 }
73
74 #header {
75 border-bottom: 1px solid #e4e4e4;
76 margin-bottom: 2rem;
77 display: flex;
78
79 position: fixed;
80 top: 0;
81 left: 0;
82 z-index: 100;
83 background-color: var(--body-color);
84 width: 100%;
85
86
87 nav {
88 height: var(--header-height);
89 display: flex;
90 align-items: center;
91 justify-content: space-between;
92 width: 100%;
93 }
94
95 .logo {
96 font: 700 1.31rem 'Poppins', sans-serif;
97 color: var(--title-color);
98 }
99
100 .logo span {
101 color: var(--base-color);
102 }
103
104 main {
105 margin-top: var(--header-height);
106 }
107
108
109 input[type=text], input[type=password] {
110 width: 100%;
111 padding: 12px 20px;
112 margin: 8px 0;
113 display: inline-block;
114 border: 1px solid #ccc;
115 box-sizing: border-box;
116 }
117
118 button {
119 background-color: var(--base-color);
120 color: var(--text-color-light);
121 height: 3.5rem;
122 display: inline-flex;
123 align-items: center;
124 padding: 0 2rem;
125 border-radius: 0.25rem;
126 font: 500 1rem 'DM Sans', sans-serif;
127 transition: background 0.3s;
128 }

```

Fonte: Autoria própria, 2022.

Figura 4 - Cadastro do Usuário Estilizado, Página Web

GetJob.

Email
123.456@gmail.com

Senha

Mostrar senha

Confirmar senha
1234

Mostrar senha

Fonte: Autoria própria, 2022.

2.3 JavaScript

É uma linguagem de programação aplicada, principalmente, para desenvolvimento web e desenvolvimento de *software*. É uma linguagem formal que funciona por meio de uma série de instruções, símbolos, palavras-chave, regras semânticas e sintáticas, de implementação de um código fonte, que permite um programador criar programas

a partir de um conjunto de ordens, ações consecutivas, dados e algoritmos que informará instruções de processamento ao computador. (PRESCOTT, 2016).

Sendo mais específico, o JavaScript (JS), atua mais na programação *front-end*, que é a parte visual de uma aplicação, mas também pode ser usado no *back-end* de aplicações, como o processamento de informações de um Banco de Dados, por exemplo.

Criada pela Netscape em parceria com a Sun Microsystems, com a finalidade de fornecer um meio de adicionar interatividade a uma página web. A primeira versão, denominada JavaScript 1.0, foi lançada em 1995 e implementada em março de 1996 no navegador *Netscape Navigator 2.0* quando o mercado era dominado pela Netscape. (SILVA, 2010).

Em conformidade com Flanagan (2011), JavaScript faz parte da tríade de tecnologias que todos os desenvolvedores Web devem conhecer: HTML, para especificar o conteúdo de páginas Web; CSS, para especificar a apresentação dessas páginas; e JavaScript, para especificar o comportamento delas. O JS é uma linguagem de alto nível, dinâmica, interpretada e não tipada, conveniente para estilos de programação orientados a objetos e funcionais.

Como fundamento do JavaScript, Silva (2010) explica algumas *tags*:

- **function:** É um procedimento de JavaScript, um conjunto de instruções que executa uma tarefa ou calcula um valor. Para usar function, você deve defini-la como quiser chamá-la.
- **const:** A declaração const cria uma variável, que é uma forma de salvar dados que possam ser acessados e/ou manipulados posteriormente, cujo valor é fixo, ou seja, somente leitura. Isso não significa que o valor é permanente, apenas que a variável constante não pode ser alterada ou retribuída.
- **value:** Retorna os valores das propriedades encontradas.
- **if:** É uma estrutura condicional que executa a afirmação, dentro do bloco, se determinada condição for verdadeira.
- **else:** Serve basicamente para suprir a parte negativa da condição do IF, ou seja, quando a condição no IF for falsa, o ELSE assume e executa a ação.
- **return:** A declaração return finaliza a execução de uma função e especifica os valores que devem ser retornados para onde a função foi chamada.

- alert: É o código utilizado para receber uma string, mensagem que será exibida ao usuário.
- var: declara uma variável, opcionalmente é possível atribuir a ela um valor em sua inicialização.

Para exemplificação, foi criada uma página de cadastro funcional feita com JavaScript. Na Figura 5 há o código da página, feito a partir das tags mencionadas:

Figura 5 - Código JavaScript

```

<script>
function esconder1() {
var esconder = document.getElementById('senha')
if (esconder.type === 'password') {
esconder.type = 'text'
} else {
esconder.type = 'password'
}
}

function esconder2() {
var esconder = document.getElementById('senha2')
if (esconder.type === 'password') {
esconder.type = 'text'
} else {
esconder.type = 'password'
}
}

function cadastrar() {
var nome = document.getElementById('nome').value;
var email = document.getElementById('email').value;
var senha = document.getElementById('senha').value;
var senha2 = document.getElementById('senha2').value;

if (senha == senha2) {
createUserWithEmailAndPassword(auth, email, senha)

alert('Usuário criado!');
} else {
return alert("Senha não condizem")
}
}
}
</script>

```

Fonte: Autoria própria, 2022.

Na Figura 8, é visível que para o cadastro ser realizado, é necessário que as senhas sejam idênticas. Assim como nas Figuras 6 e 7.

Figura 6 - Cadastro do Usuário Funcional, Página Web

The screenshot shows a web browser window with the address bar displaying 'Arquivo | E:/Etec/cadastropr1.html'. The page title is 'Cadastro'. The form contains the following elements:

- Email:** A text input field containing 'ds@etec.sp.gov.br'.
- Senha:** A password input field with masked characters (dots) and a checkbox labeled 'Mostrar senha'.
- Confirmar senha:** A second password input field with masked characters and a checkbox labeled 'Mostrar senha'.
- Botão:** A button labeled 'Cadastro' at the bottom left of the form.

Fonte: Autoria própria, 2022.

Figura 7 - Cadastro do Usuário Funcional, Página Web (2)

The screenshot shows a web browser window with the address bar displaying 'Arquivo | E:/Etec/cadastropr1.html'. The page title is 'Cadastro'. The form contains the following elements:

- Email:** ds@etec.sp.gov.br
- Senha:** 12345678, with a checked checkbox for 'Mostrar senha'.
- Confirmar senha:** 12345678, with a checked checkbox for 'Mostrar senha'.
- Button:** Cadastro

Fonte: Autoria própria, 2022.

Figura 8 - Cadastro do Usuário Funcional, Página Web (3)

The screenshot shows the same web browser window as in Figure 7, but with a validation error message displayed in a white box on the right side of the form. The message reads: 'Essa página diz Senhas batem' (This page says Passwords match). There is a blue 'OK' button next to the message. The form fields and buttons remain the same as in the previous screenshot.

Fonte: Autoria própria, 2022.

2.4 Banco de Dados

Como dito por Ramakrishnan e Gehrke (2011), bancos de dados são coleções de dados que descrevem como as relações organizadas estão funcionando. Sendo assim, um sistema de gerenciamento de banco de dados (SGBD), virou um *software* para o auxílio e manutenção para vastos conjuntos de dados. O que tornou os sistemas de gerenciamento de banco de dados uma das ferramentas mais importantes para o gerenciamento de dados ou informações. E a disciplina dos princípios práticos dos sistemas de banco de dados tornou-se uma parte inerente do currículo de Ciência da Computação.

Um sistema de banco de dados é basicamente apenas um sistema computadorizado de manutenção de registros. O banco de dados, por si só, pode ser considerado como o equivalente eletrônico de um armário de arquivamento; ou seja, ele é um repositório ou recipiente para uma coleção de arquivos de dados computadorizados. (Date, 2003, Pg. 3).

Figura 9 - Organização de Dados

Fonte: Autoria própria, 2022.

Dessa forma, de acordo com Mara e Cardoso (2012), quando pensamos em projetar um banco de dados, devemos pensar em como serão armazenados os dados, de início, começamos pensando como será a estrutura em que as informações serão armazenadas, e como funcionará sua segurança.

Pelas palavras de Elmasri e Navathe (2005), os bancos de dados multimídias são atualmente capazes até de guardar figuras, áudios e vídeos. Ou também bancos de sistemas de informações geográficas chamados de *geographic information system* (GIS), onde analisam mapas em tempo real por satélites; ou também o banco *online analytical processing* (OLAP), onde ele faz um processamento analítico online que é usado para extrair informações no banco na hora de tomar decisões.

2.4.1 Banco de Dados não relacional

Como dito por Sadalage e Fowler (2013), desde que se começou o uso de banco de dados, os bancos relacionais têm sido a escolha principal para o armazenamento vários dados, principalmente quando nos referimos a aplicativos corporativos. Onde, por muito tempo, esse tipo de banco dominou o mercado como escolha de banco de dados na década 90. Porém, após um longo tempo de domínio de mercado, surgem os bancos de dados *Not Only Standard Query Language* (NoSQL), bancos não relacionais, sendo esses bancos de dados que atuam sem um esquema definido, permitindo que sejam livremente adicionados campos aos registros dos dados no banco, sem que seja necessário primeiramente definir quaisquer mudanças na estrutura.

Sempre que um time inicia um novo projeto, uma das decisões mais relevantes é qual linguagem será usada para a programação, os principais frameworks e como será a arquitetura. Entretanto, Paniz (2016), expôs que algo que é pouco reforçado, porém

de suma importância, é como será a persistência dos dados. Sendo que, uma das soluções atuais para essa persistência de dados, são os bancos NoSQL, que são uma forma moderna para o armazenamento de dados.

Oliveira, Pontes e Lóscio (2011), afirmam que os grandes volumes de dados sendo gerados em aplicações web, juntamente com os requisitos diferenciados destas aplicações, como por exemplo a escalabilidade e o grau de disponibilidade, fizeram com que os bancos NoSQL virassem uma opção, sendo essa uma proposta de banco que permite o gerenciamento de grandes volumes de dados semiestruturados ou não estruturados, que permitam altas demandas de escalabilidade e disponibilidade.

Já dizia Paniz (2016), quando se vai escolher um banco não relacional, existem várias opções de bancos para isso, sendo que normalmente o método em que o banco funciona normalmente já é explícito em seu nome, sendo os principais tipos:

- Chave-Valor: um método onde todos os registros fazem parte de uma mesma coleção de elementos, com a única coisa em comum entre eles sendo sua chave única;
- Colunar: neste caso, é um método em que todos os registros fazem parte de uma mesma tabela, com a diferença em que eles podem ter colunas diferentes;
- Documento: já neste, os registros ficam armazenados em uma coleção específica, sendo que não necessariamente é preciso que todos eles fiquem no mesmo esquema fixo;
- Grafo: por último, os sistemas em grafo, que são sistemas ligados por alguns relacionamentos.

2.4.2 Normalização

O conceito de normalização é aplicado ao projeto de Banco de Dados após a finalização do modelo do próprio. Graças a normalização, erros como a repetição de dados, Chave Primária concatenadas, perda de informação e dificuldade de representar um fato e/ou objeto real não prejudicam o funcionamento do sistema, pois são excluídos no processo. Tal conceito foi inserido por Edgar Frank Codd, em 1970, através de fundamentos na teoria matemática de conjuntos.

Quando é observado que uma tabela não está normalizada (ÑN) em um Banco de Dados, é preciso formatá-la para que se iguale a Primeira Forma Normal (1FN), assim,

evitando a repetição de dados. A 1FN requer que todas as colunas de uma tabela possuam valores atômicos. Por exemplo, grupos repetidos são eliminados pela normalização, pois ela os põe em tabelas separadas através da conexão com chaves primárias ou estrangeiras.

Na Figura 10, há o exemplo da Tabela Cliente, que está ÑN, onde é observado mais de um número de telefone na mesma linha. Além de informações como Rua, Bairro e CEP também estarem na mesma linha:

Figura 10 - Tabela Cliente ÑN

Código_cliente	Nome	Telefone	Endereço
C001	Henrique	97739-9021	Av. Liberdade, 720 Liberdade 01503-000
C002	Marcelo	97573-4000	Av. Cidade Jardim, 38 Jardins 69317-529
C003	Rose	96733-8929 96733-8930	Av. Morumbi, 45 Morumbi 05650-001

Fonte: Autoria própria, 2022.

Como primeiro passo para transformar a Tabela Cliente Não Normalizada em 1FN, as informações independentes da coluna Endereço devem ser separadas, assim como mostra a Figura 11:

Figura 11 - Tabela Cliente ÑN (2)

Código_cliente	Nome	Telefone	Rua	Bairro	CEP
C001	Henrique	97739-9021	Av. Liberdade, 720	Liberdade	01503-000
C002	Marcelo	97573-4000	Av. Cidade Jardim, 38	Jardins	69317-529
C003	Rose	96733-8929 96733-8930	Av. Morumbi, 45	Morumbi	05650-001

Fonte: Autoria própria, 2022.

Já na Figura 12, para encerrar o processo de transformação da Primeira Forma Normal, é criada a entidade Telefone, os separando da Tabela Cliente e utilizando o Código_cliente (Código do Cliente) como referência:

Figura 12 - Tabela Cliente e Telefone Normalizadas na 1FN

Código_cliente	Nome	Rua	Bairro	CEP
C001	Henrique	Av. Liberdade, 720	Liberdade	01503-000
C002	Marcelo	Av. Cidade Jardim, 38	Jardins	69317-529
C003	Rose	Av. Morumbi, 45	Morumbi	05650-001

Código_cliente	Telefone
C001	97739-9021
C002	97573-4000
C003	96733-8929
C003	96733-8930

Fonte: Autoria própria, 2022.

Para uma relação estar na Segunda Forma Normal (2FN), ela necessariamente deve estar na 1FN. Cada atributo não-chave não poderá ser dependente de apenas parte da chave, portanto, cada atributo não-chave não poderá ter qualquer tipo de dependência funcional ou não-trivial. Desta maneira, a 2FN evita inconsistências devido a duplicidades.

Analisando a Tabela Venda Não Normalizada, Figura 13, é visível que o nome do produto depende do código do produto, entretanto, não depende do Número do pedido, que é a Chave Primária da tabela.

Figura 13 - Tabela Venda Não Normalizada na 2FN

N_pedido	Código_produto	Produto	Quant	Valor_unit	Subtotal
501	1-923	Placa de Vídeo	3	2.000,00	6.000,00
502	1-989	Placa Mãe	5	1.000,00	5.000,00
503	1-976	Monitor	2	1.500,00	3.000,00
504	1-944	SSD	8	300,00	2.400,00

Fonte: Autoria própria, 2022.

Para que seja normalizada, deve ser criada a entidade Produto, a partir dos atributos Código_produto (Código do produto) e Produto (Nome do produto), como na Figura 14:

Figura 14 - Tabela Venda e Produto Normalizadas na 2FN

N_pedido	Código_produto	Quant	Valor_unit	Subtotal
501	1-923	3	2.000,00	6.000,00
502	1-989	5	1.000,00	5.000,00
503	1-976	2	1.500,00	3.000,00
504	1-944	8	300,00	2.400,00

Código_produto	Produto
1-923	Placa de Vídeo
1-989	Placa Mãe
1-976	Monitor
1-944	SDD

Fonte: Autoria própria, 2022.

A Terceira Forma Normal (3FN) exige que a relação esteja previamente na 2FN, e que atributos não-chave não façam referência a outros atributos não-chave. A coluna que não depende diretamente da chave será retirada e em seguida será criada uma relação que conterà determinado grupo de atributos.

Na Tabela Venda ÑN, Figura 15, vemos que a coluna Subtotal depende das colunas Quant (Quantidade) X Valor_unit (Valor da unidade), logo, não está na 3FN. Desse modo, para normalizar a tabela, a coluna Subtotal será eliminada, como mostra a Figura 16:

Figura 15 - Tabela Venda Não Normalizada na 3FN

N_pedido	Código_produto	Quant	Valor_unit	Subtotal
501	1-923	3	2.000,00	6.000,00
502	1-989	5	1.000,00	5.000,00
503	1-976	2	1.500,00	3.000,00
504	1-944	8	300,00	2.400,00

Fonte: Autoria própria, 2022.

Figura 16 - Tabela Venda Normalizada na 3FN

N_pedido	Código_produto	Quant	Valor_unit
501	1-923	3	2.000,00
502	1-989	5	1.000,00
503	1-976	2	1.500,00
504	1-944	8	300,00

Fonte: Autoria própria, 2022.

2.4.3 Dicionário de Dados

O Dicionário de Dados (DD) é onde encontram-se os elementos de dados que são importantes para o sistema. Um Dicionário de Dados descreve entradas, saídas, composição de depósitos de dados e alguns cálculos intermédios. Melhor dizendo, é uma lista com todos os elementos envolvidos e seus significados.

Figura 17 - Símbolos do Dicionário de Dados

Símbolo	Significado
=	é constituído por ou é definido por
+	e (conjunção ou concatenação)
()	enquadram componentes opcionais
[]	enquadram componentes oque são utilizadas alternativamente
	separam componentes alternativas enquadradas por []
{ }	enquadram componentes que se repetem 0 ou mais vezes
**	enquadram comentários
@	identifica a chave primária de um depósito

Fonte: Autoria própria, 2022.

Como exemplo, a Figura 18 retrata a estrutura do Dicionário de Dados da tabela Venda:

Figura 18 - Exemplo de Dicionário de Dados

Venda = (@N_pedido; Código_produto; Quant; Valor_unit)
N_pedido = {Digitos} *Chave Primária*
Código_produto = {Digitos} *Chave estrangeira da entidade*
Quant = {Digitos} *Valor de cada unidade*
Valor_unit = {Digitos} *Valor de cada unidade*
Digitos = [0 1 2 3 4 5 6 7 8 9]

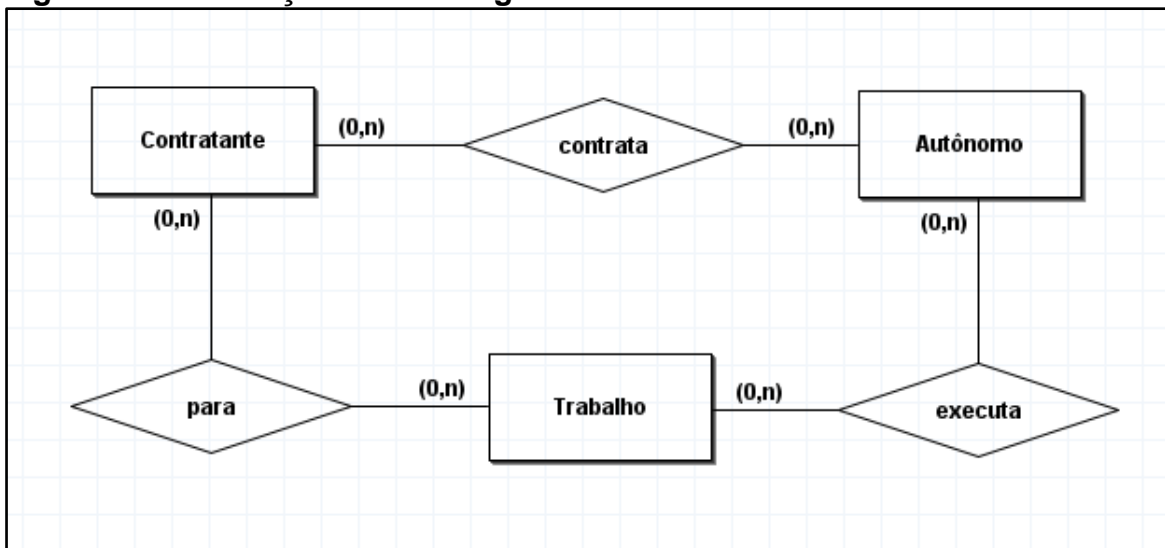
Fonte: Autoria própria, 2022.

2.4.4 Diagramas ER

Segundo Ramakrishnan e Gehrke (2011), os diagramas ER são uma descrição de dados aproximados numa base de como o site irá funcionar, sendo que a mesma é feita em forma de uma avaliação subjetiva com base nos dados dos requisitos. Para eles, o modelo ER é o modelo mais relevante numa criação de um banco de dados.

Sendo uma técnica para representação de uma estrutura lógica de como ficara o banco de dados, o diagrama ER é um meio para a simplificação do entendimento da estrutura lógica geral do banco e como ele irá se comunicar com os principais aspectos do projeto do banco. Pois para alguém trabalhar com o banco, um meio visual simulando todas as conexões do projeto auxilia muito seu entendimento. (DATE, 2003).

Figura 19 - Simulação de um diagrama ER



Fonte: autoria própria, 2022.

Na Figura 19, é simulado visualmente como seria a contratação de um autônomo que realiza um trabalho para o contratante. Dessa mesma forma seria o diagrama ER, que simula visualmente como será a funcionalidade do banco para o programador. É possível identificar alguns retângulos, que são as entidades, sendo elas usadas para registrar certos atributos e chaves de identificação para o banco. Enquanto o losango representando o relacionamento, que poderia ser chamado de “entidade especial”, onde, ao contrário de uma condição, eles medem os conceitos de duas entidades para que sejam ligadas de alguma maneira. (DATE, 2003).

2.5 Firebase

De acordo com Machado (2021), um dos principais problemas que se tem em qualquer *software*, é na parte da criação, organização e manutenção dos dados. E é exatamente com esse problema que o Firebase entra em ação, sendo uma infraestrutura recheado com vários recursos e permitindo qualidade e escalabilidade. Este serviço fornecido pelo Google permite que os desenvolvedores possam direcionar os seus esforços para outras partes do projeto, como na interface, ou deixar

ela mais engajado para os usuários, deixando para que o Google dimensione automaticamente a infraestrutura.

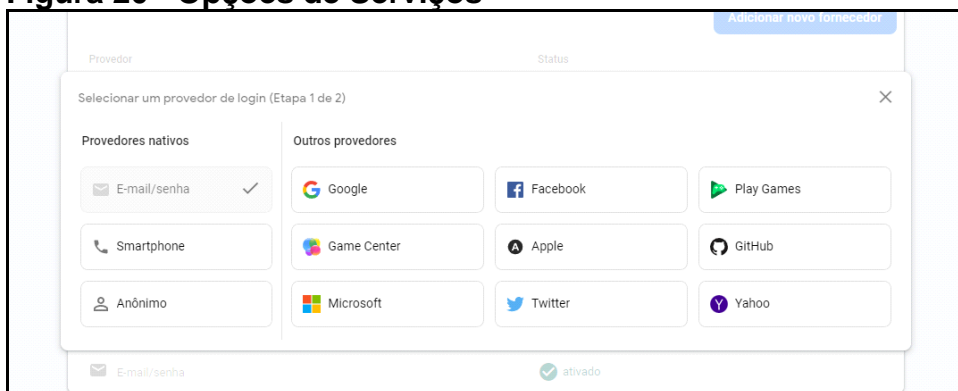
Dessa forma, com o Firebase, é possível criar aplicativos avançados, ativar o *back-end* sem gerenciar servidores e ainda escalonar o projeto para que seja possível atender a milhares de usuários, por meio das funções do banco de dados, além de dar acesso a *machine learning*, e soluções para armazenamento de grandes dados, a partir do *cloud function* e uma solução de hospedagem.

2.5.1 Funções do Firebase

Firebase Authentication: Uma das partes mais complexas no desenvolvimento do banco de dados, certamente é a forma de autenticação, pois fazer uma que seja segura e boa, leva tempo e recursos. Porém, como uma solução a isso, o Firebase fornece o *Authentication*, que é uma solução segura e rápida para esses problemas, assim, proporcionando uma experiência positiva aos usuários. (MACHADO, 2021).

Por meio de serviços de *back-end* e de *Software Development Kits (SDKs)*, o serviço de autenticação do google permite que além de um sistema de autenticação, ele também de suporte ao sistema de login, utilizando como base outros provedores de identidade populares, como o Google, Twitter, entre outros.

Figura 20 - Opções de Serviços



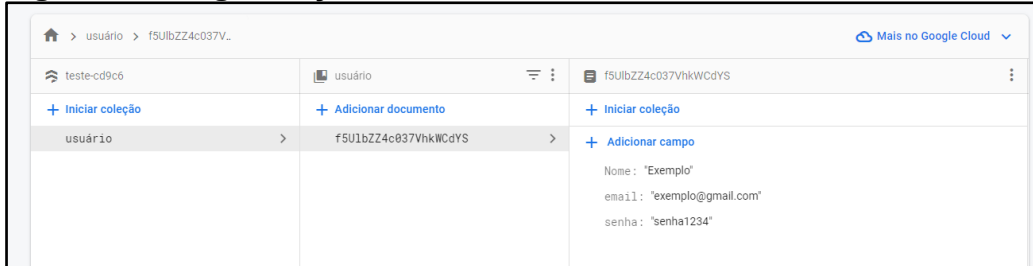
Fonte: Autoria própria, 2022.

Cloud Firestore: Este é o serviço do Firebase que permite o uso de dados de nuvem, por meio do NoSQL flexível e escalável, com armazenagem e sincronização de dados em tempo real, usando só servidores Google.

Dessa maneira, o Firestore se tornou uma maneira simples para que desenvolvedores possam fazer bancos de dados escalonáveis, com suporte a vários tipos de projetos, como web ou dispositivos móveis, através do *Google Cloud*. Permitindo que os dados

sejam atualizados em tempo real para os clientes, e assim armazenar grandes quantidades de dados.

Figura 21 - Organização em Pastas

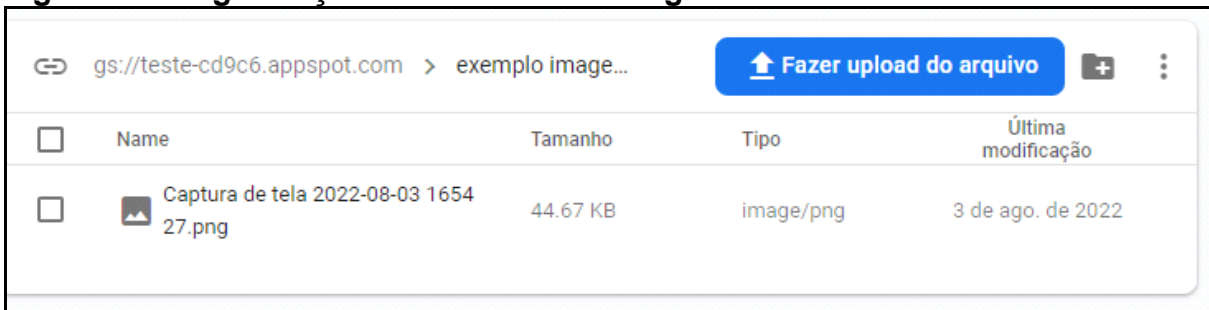


Fonte: Autoria própria, 2022.

Firestore: O *Cloud Storage* foi um serviço feito pelo Google, com o objetivo de que desenvolvedores pudessem armazenar e exibir imagens ou vídeos que os usuários enviassem.

Assim, o *Firestore* se tornou um meio útil e avançado para que se possa implementar o armazenamento de vários dados não textuais no banco de dados, como imagens e vídeos, ainda usufruindo da segurança do Google, e ainda funcionar independentemente da qualidade de internet do usuário.

Figura 22 - Organização de Pastas no Storage



Fonte: Autoria própria, 2022.

Firestore: Sendo feito para projetos web, o *Firestore* é o serviço do Firebase que permite hospedar os sites na internet de forma simples. Assim, permitindo a criação de aplicações web de forma dinâmica, com distribuição global, operando com os servidores da Google.

Figura 23 - Hospedar Projeto no Firebase

Fonte: Autoria própria, 2022.

2.6 UML

A Unified Modeling Language (UML), Linguagem de Modelagem Unificada, teve sua origem com base na união de três metodologias; o método de Booch, o método *Object Modeling Technique* (OMT) e o método *Object-Oriented Software Engineerin* (OOSE). Sendo que esses eram os métodos mais populares em meados de 1990, a *Rational Software Process* (RUP) começou com uma proposta da unificação dos três métodos e, a partir desta proposta, se data o começo da criação do UML. Onde, primeiramente, com uma junção do método Booch com o método OMT, nos anos de 1995, se deu a origem ao Método Unificado; que no ano seguinte, em 1996, teria o acréscimo do método OOSE, o que fez com que renomeassem UML. (GUEDES, 2018).

Como expressava Fowler (2014), UML é um meio de notações gráficas para o auxílio da construção de um projeto de *software*, principalmente para os que são orientados por objetos. Sendo assim, UML é, basicamente, um esboço do que é o projeto em um geral, resumindo como será feito a programação e como o projeto será conduzido. Apesar de que o conceito de UML possa variar de pessoa para pessoa, o princípio geral se mante como a forma na qual se organizará o projeto em esboço, o projeto em si e a linguagem de programação.

Sendo assim, na hora da criação do seu projeto, o UML tem seu uso como a linguagem gráfica para a visualização, especificação, construção e documentação das funções. Posto isso, sua necessidade se vem devido à importância de proporcionar um formato padrão na preparação da estrutura do sistema, desde os conceitos fora da programação como as regras de negócio, ou diretamente nas funções do sistema como determinar a linguagem na programação, quais serão os conteúdos reutilizados do *software* e até como ocorrerá os esquemas do banco de dados. (BOOCH e RUMBAUGH, 2006).

No entanto, como mencionado por Guedes (2018), a UML, apesar de ser uma linguagem de modelagem, ela não é propriamente uma linguagem de programação, mas sim, uma linguagem visual para a modelagem de *softwares*. Dessa maneira, seu propósito não é diretamente no que se refere a programação prática do site, e sim na parte da modelagem, para que assim sirva de auxílio aos engenheiros de *software*, que já terão as características do sistema, seus requisitos, estrutura, lógica e comportamento já definidos antes de começar a programação interna, permitindo uma abordagem mais organizada e fácil do projeto.

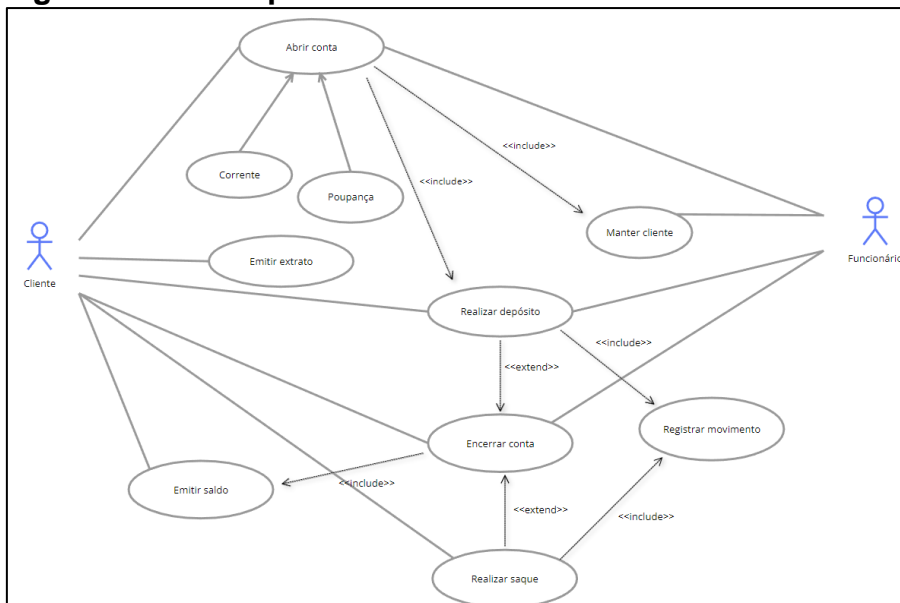
2.6.1 Diagrama de Casos de Uso

O diagrama de caso de uso documenta o que o sistema faz do ponto de vista do usuário. É um instrumento eficiente para determinação e documentação dos serviços a serem desempenhados pelo sistema. Em outras palavras, ele descreve as principais funcionalidades e a interação dessas com os usuários do mesmo sistema. É a representação das funcionalidades externamente observáveis e dos elementos externos ao sistema e que com ele interagem. (BEZERRA, 2007).

Segundo Bezerra (2007), um caso de uso não é um passo em uma funcionalidade do sistema. Ao contrário, um caso de uso é um relato fim a fim de um dos usos do sistema por um agente externo.

Para fins de exemplificar, na Figura 24, um diagrama de caso de uso de um controle bancário:

Figura 24 – Exemplo de Sistema de Controle Bancário

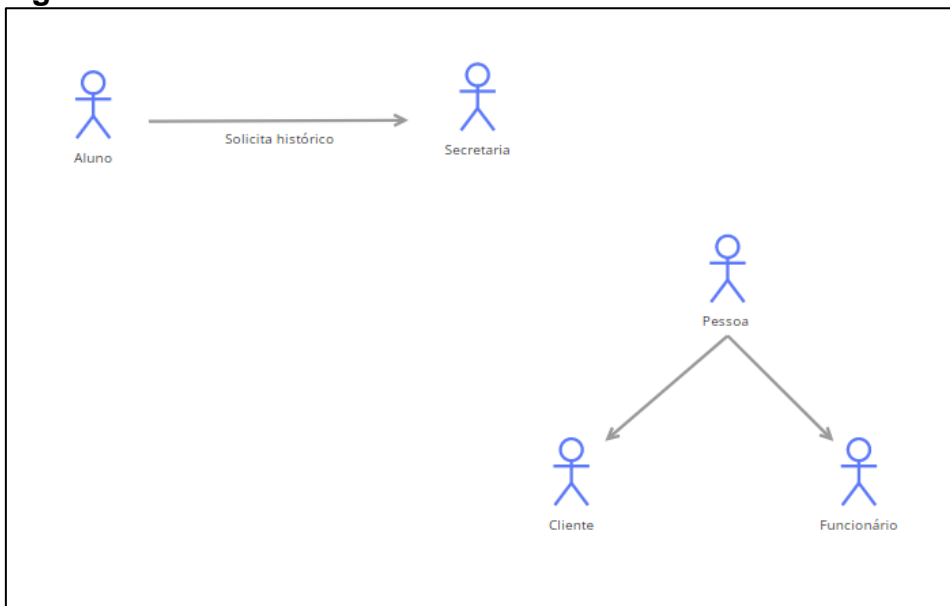


Fonte: Autoria Própria, 2022.

Dentro do diagrama possuem os atores, que são entidades externas ao sistema que está sendo modelado, as entidades externas representadas pelos atores podem ser: pessoas, dispositivos, *hardwares* ou *softwares*.

Os relacionamentos entre atores podem ser desprezados, mas caso o analista os veja como necessários ou úteis, então, ele poderá incluí-los no modelo de casos de uso. São apresentados exemplos de atores, como “Aluno”, “Secretaria”, “Cliente”, “Pessoa” e “Funcionário”, e a ligação entre eles na Figura 25:

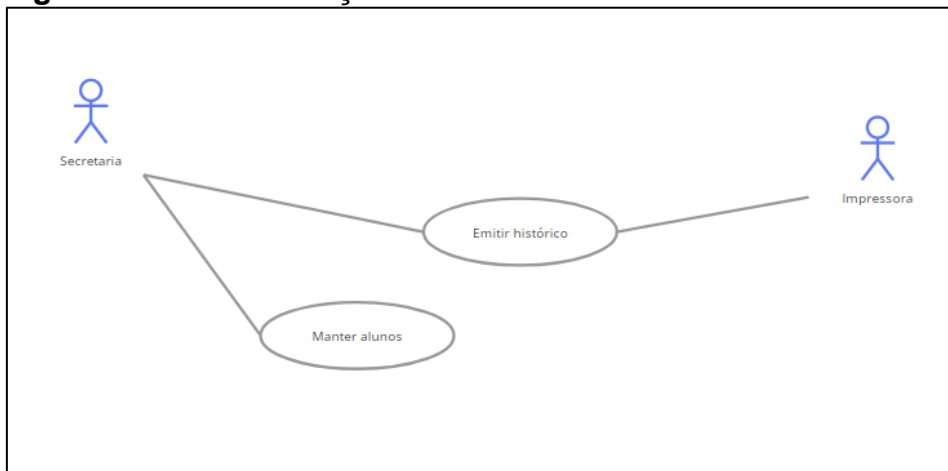
Figura 25 – Relacionamento Entre os Atores



Fonte: Autoria própria, 2022.

Durante um relacionamento entre um ator e um caso de uso, é representada uma comunicação, pois o ator sendo uma entidade externa não pode possuir qualquer relação estrutural com o sistema computacional. A notação UML para este tipo de relacionamento é um segmento de reta ligando ator e caso de uso. (STADZISZ, 2002).

Um exemplo, na Figura 26, da comunicação entre atores:

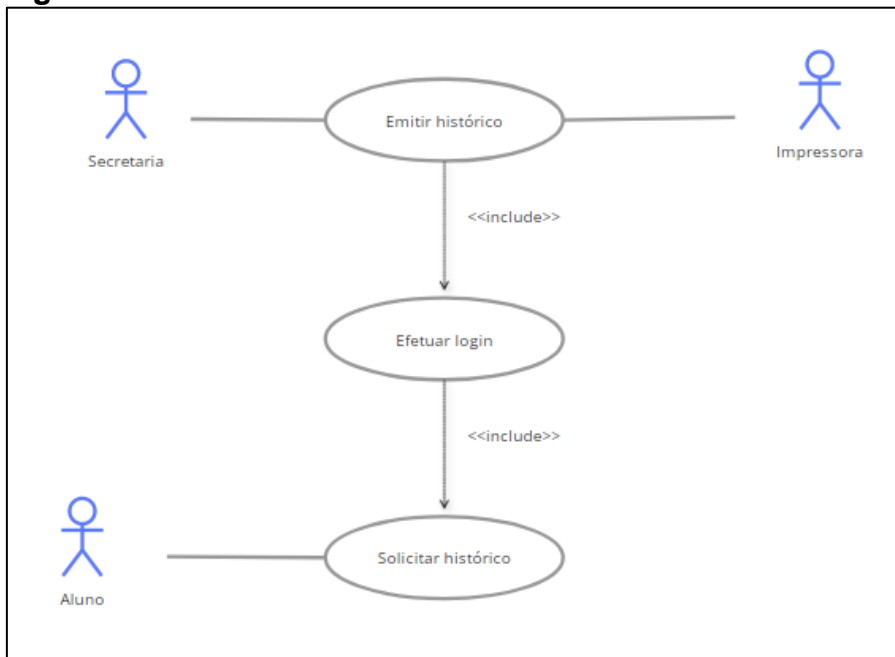
Figura 26 – Comunicação Entre Atores e Casos de Uso

Fonte: Autoria Própria, 2022.

As relações entre casos de uso nunca serão do tipo comunicação. Isto ocorre porque casos de uso são aplicações completas do sistema e, por consequência, não existe sentido em fazer comunicar dois “usos do sistema”. Todas as relações entre casos de uso serão sempre estruturais. Existem três tipos de relações / associações entre casos de uso: inclusão, extensão e generalização. (STADZISZ, 2002).

O princípio sobre relacionamento de inclusão entre casos de uso é o mesmo utilizado no mecanismo de definição de funções em linguagens de programação. Segundo Bezerra (2007), quando dois ou mais casos de uso incluem uma sequência comum de interações, essa sequência comum pode ser descrita em outro caso de uso. Partindo disso, vários casos de uso do sistema podem incluir o comportamento desse caso de uso comum. Um exemplo de relacionamento de inclusão na Figura 27:

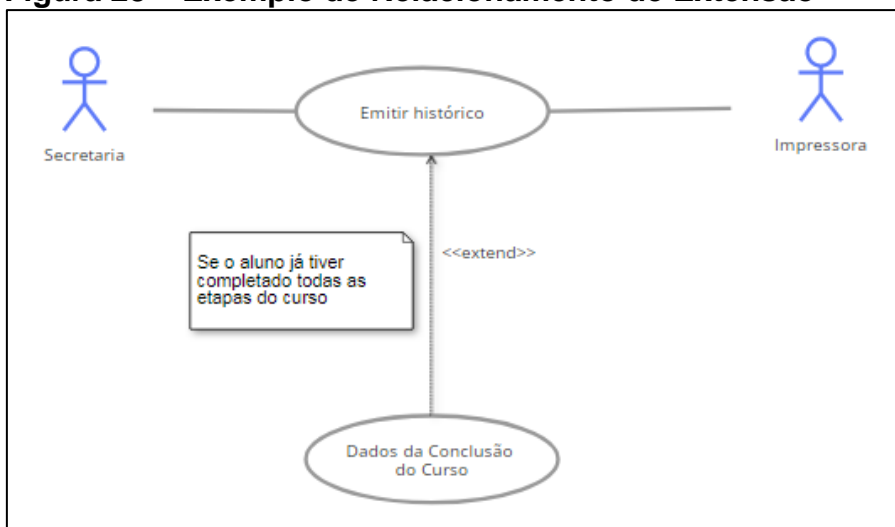
Figura 27 - Relacionamento de Inclusão



Fonte: Autoria Própria, 2022.

Um relacionamento de extensão é uma relação estrutural entre dois casos de uso, através da qual um caso de uso maior é estendido por um caso de uso menor. A extensão significa que o caso de uso que estende, inclui serviços especiais em um caso de uso maior. Com a Figura 28, podemos observar um exemplo de relacionamento de extensão:

Figura 28 – Exemplo de Relacionamento de Extensão



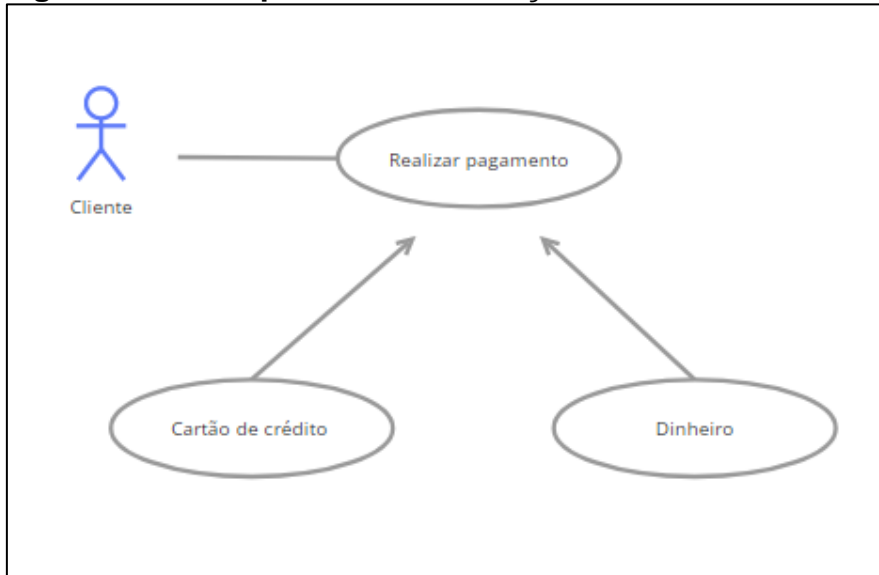
Fonte: Autoria Própria, 2022.

A generalização mostra variações específicas em um caso de uso mais geral, de forma análoga à generalização entre classes. Um caso de uso mãe representa uma sequência geral de comportamentos; os casos de uso filhos especializam o caso de

uso mãe, inserindo etapas adicionais ou detalhando etapas. (BLAHA e RUMBAUGH, 2006).

Um exemplo de generalização, Figura 29:

Figura 29 - Exemplo de Generalização

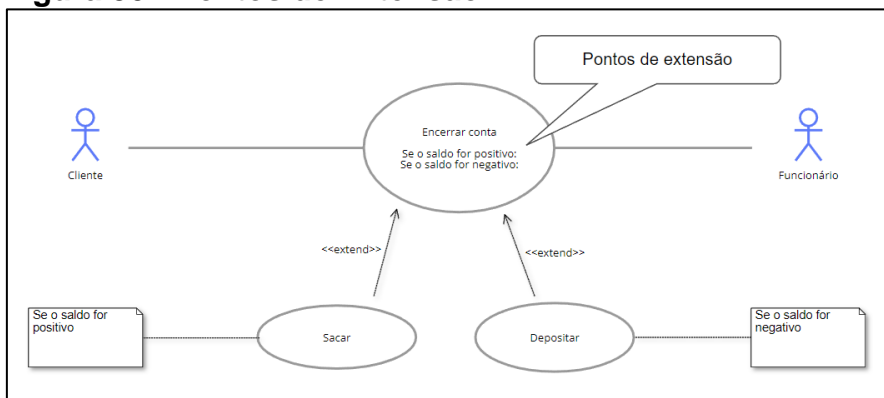


Fonte: Autoria Própria, 2022.

Deve ser colocada uma nota explicativa ou comentário no diagrama sempre que necessário. É utilizada para explicar as condições, para que relacionamentos de extensão sejam realizados. Possuem o intuito de ajudar no entendimento do que se quer expressar no diagrama UML.

Pode ser utilizado também um ponto de extensão, que identifica um ponto no comportamento de um caso de uso a partir do qual esse comportamento poderá ser estendido pelo comportamento de outro caso de uso, se a condição for satisfeita. Como é exemplificado na Figura 30:

Figura 30 - Pontos de Extensão



Fonte: Autoria Própria, 2022.

Como mencionado por Guedes (2011), os diagramas de caso de uso costumam ser documentados por instruções em linhas gerais de como será o funcionamento; quais atividades deverão ser executadas; qual evento forçará sua execução; quais atores poderão utilizá-los e quais suas possíveis restrições.

A seguir, na Figura 31, um exemplo de documentação de caso de uso:

Figura 31 - Exemplo de Documentação de Caso de Uso

Nome do Caso de Uso	Abrir Conta
Caso de Uso Geral	
Ator Principal	Cliente
Atores Secundários	Funcionário
Resumo	Esse caso de uso descreve as etapas percorridas por um cliente para abrir uma conta corrente
Pré-Condições	O pedido de abertura precisa ter sido previamente aprovado
Pós-Condições	É necessário realizar um depósito inicial
Fluxo Principal	
Ações do Ator	Ações do Sistema
1. Solicitar Abertura de Conta	
	2. Consultar cliente por seu CPF ou CNPJ
3. Informar a senha da conta	
	4. Abrir conta
5. Fornecer valor a ser depositado	
	6. Registrar depósito
	7. Emitir cartão da conta

Fonte: (Guedes, 2006).

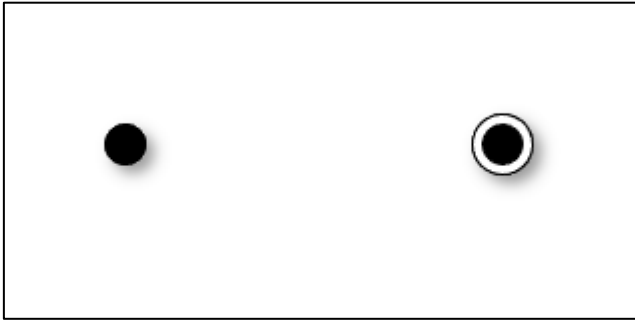
2.6.2 Diagrama de Atividades

É um diagrama criado para modelagem de processos. Elabora os fluxos de controle, adicionando decisão e paralelismo, uma ferramenta útil para modelar processos, como casos de uso e passos de execução de uma rotina. (FOWLER, 2005).

No diagrama de atividades, o objetivo é modelar um processo em que pode envolver diversos objetos e implicar diferentes estados nos objetos.

Segundo Pressman (2021), são usados símbolos de início e fim para representar atividades e ações. As ações são procedimentos atômicos que não podem ser decompostos e as atividades são procedimentos compostos que têm um sub diagrama de atividades próprio. Uma ação pode ser posteriormente detalhada, se tornando uma atividade.

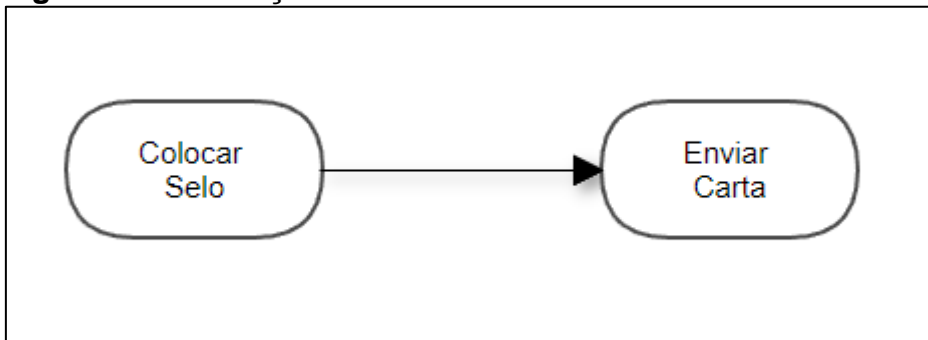
Figura 32 - Símbolos de início e fim



Fonte: Autoria própria, 2022.

Os fluxos de controle são usados para determinar precedência entre atividades. Uma atividade só pode executar após a execução de todas as atividades com fluxos que desemboquem nela. Como exemplo, a Figura 33 demonstra como a atividade “Enviar Carta” só pode ser executada após a atividade “Colocar Carta” estiver concluída:

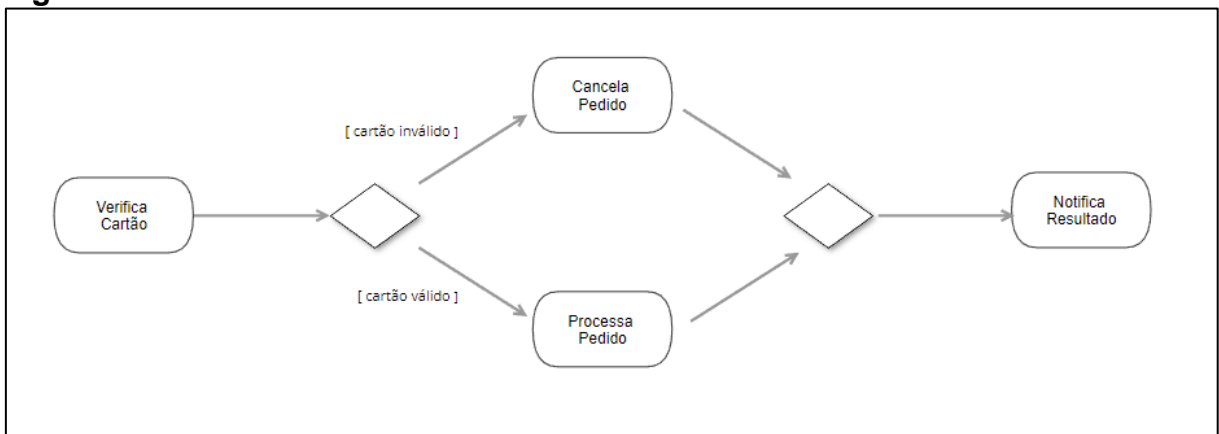
Figura 33 - Execução das atividades



Fonte: Autoria própria, 2022.

O diagrama de atividades permite que decisões sejam tomadas durante o fluxo. Os fluxos que saem das decisões têm condições de guarda, determinando qual caminho seguir. Como apresenta a Figura 34:

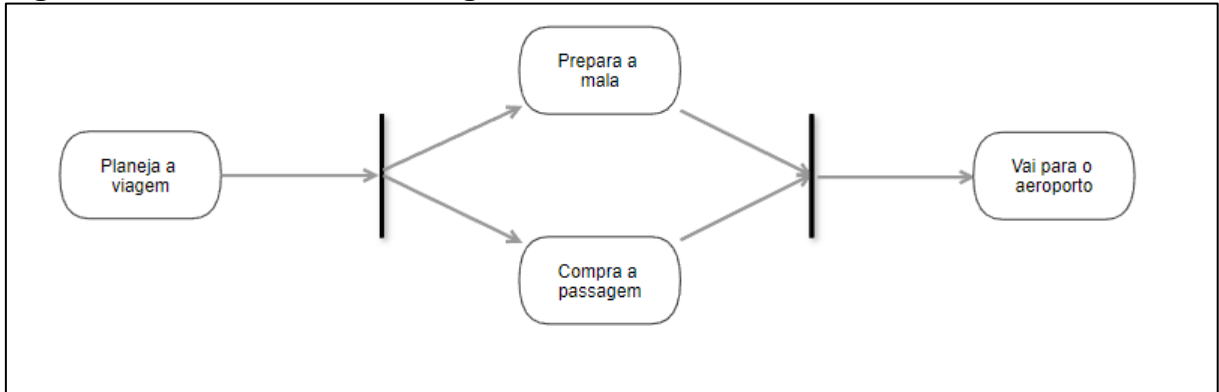
Figura 34 - Decisões Durante o Fluxo



Fonte: Autoria própria, 2022.

Existe o paralelismo, que permite que duas ou mais atividades sejam feitas em paralelo, e o sincronismo, que possibilita que sejam demarcados pontos de controle que obrigam a chegada dos fluxos para continuar a execução. Na Figura 35, um exemplo disso:

Figura 35 - Paralelismo no Diagrama de Atividades

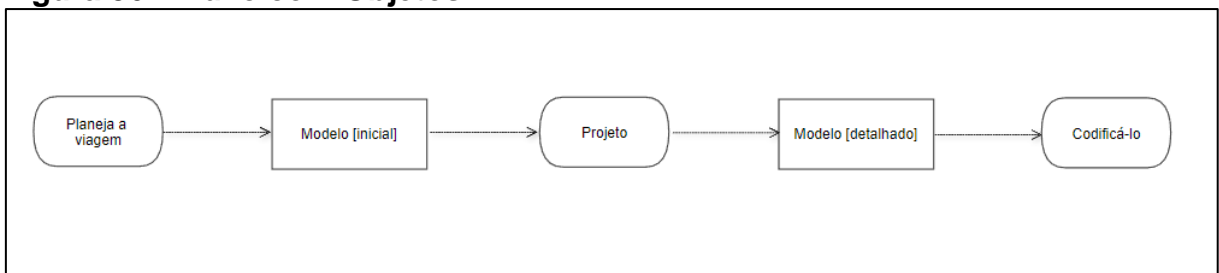


Fonte: Autoria própria, 2022.

Os objetos são produtos requeridos ou gerados pela execução de atividades. Um fluxo com objetos deixa implícito um fluxo de controle. Um objeto é descrito por nome e estado, entre colchetes. (FOWLER, 2005).

Os retângulos de canto arredondado representam atividades ou ações; as caixas representam objeto; as linhas representam fluxos de controle. Como apresenta a Figura 36:

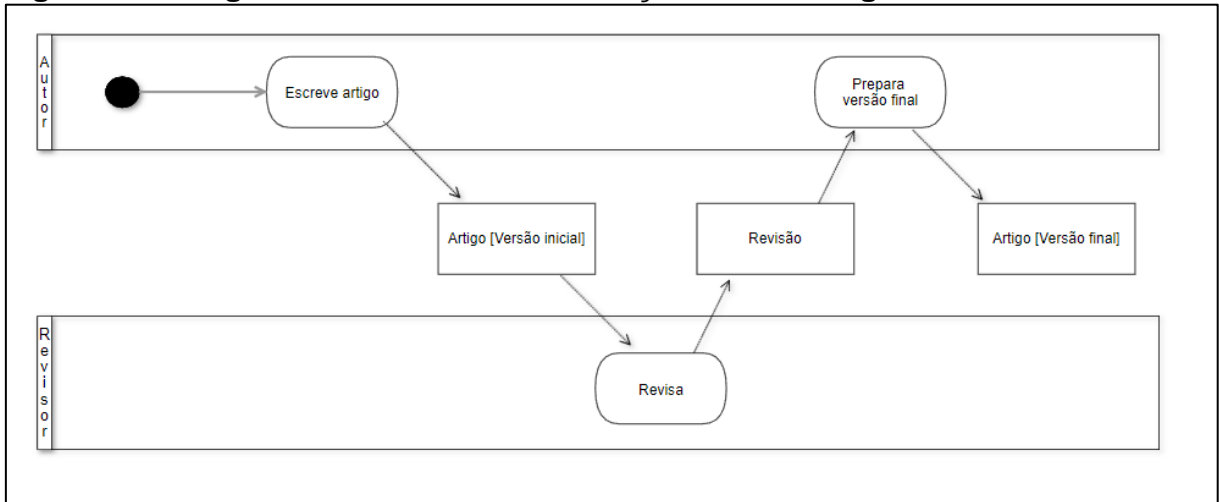
Figura 36 - Fluxo com Objetos



Fonte: Autoria própria, 2022.

Caso deseje representar os papéis responsáveis por cada atividade, pode ser utilizado o recurso de raias, que podem ser horizontais ou verticais. Da mesma forma que é mostrado no exemplo de criação de um artigo no diagrama da Figura 37:

Figura 37 - Diagrama de Atividade da Criação de um Artigo



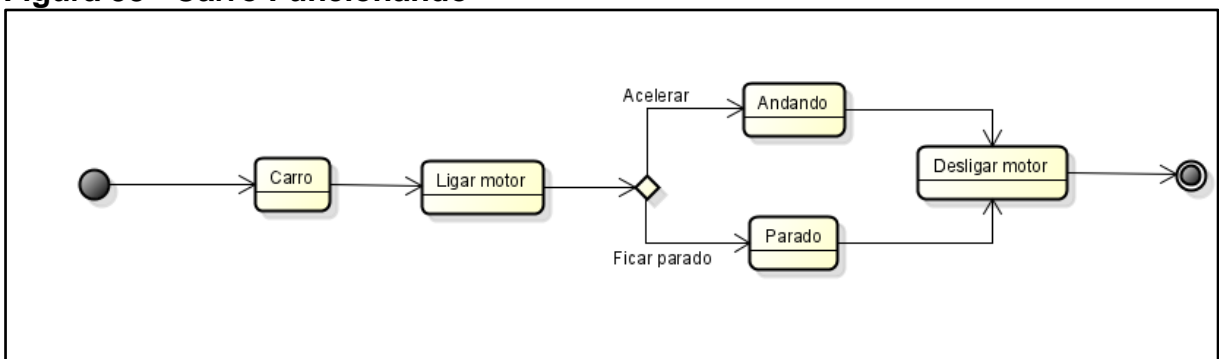
Fonte: Autoria própria, 2022.

2.6.3 Diagrama de Máquina de Estado

O diagrama de máquina de estado simula o ciclo de um objeto, reproduzindo como seria as interações deles com determinados eventos que ele possa passar, além de mostrar suas transições, e seus estados. Sendo este diagrama um método ilustrado para que se possa ter noção de todos os estados que um objeto pode passar. (CRAIG, 2000).

Com base no exemplo da Figura 38, simulando um carro ligando e desligando, o diagrama de estado representa as ações que estão acontecendo ao ligar o carro, se ele vai se movimentar e, logo após, seu desligamento.

Figura 38 - Carro Funcionando



Fonte: Autoria Própria, 2022.

O diagrama se inicia no “Carro”, a sequência de eventos, indicada pelas setas, levam o “Carro” para o estado “Ligar motor”, que então redireciona para outros dois estados, sendo o estado “Andando” e o estado “Parado”, que terminam no estado “Desligar motor”, resultando no fim do diagrama. (CRAIG, 2000).

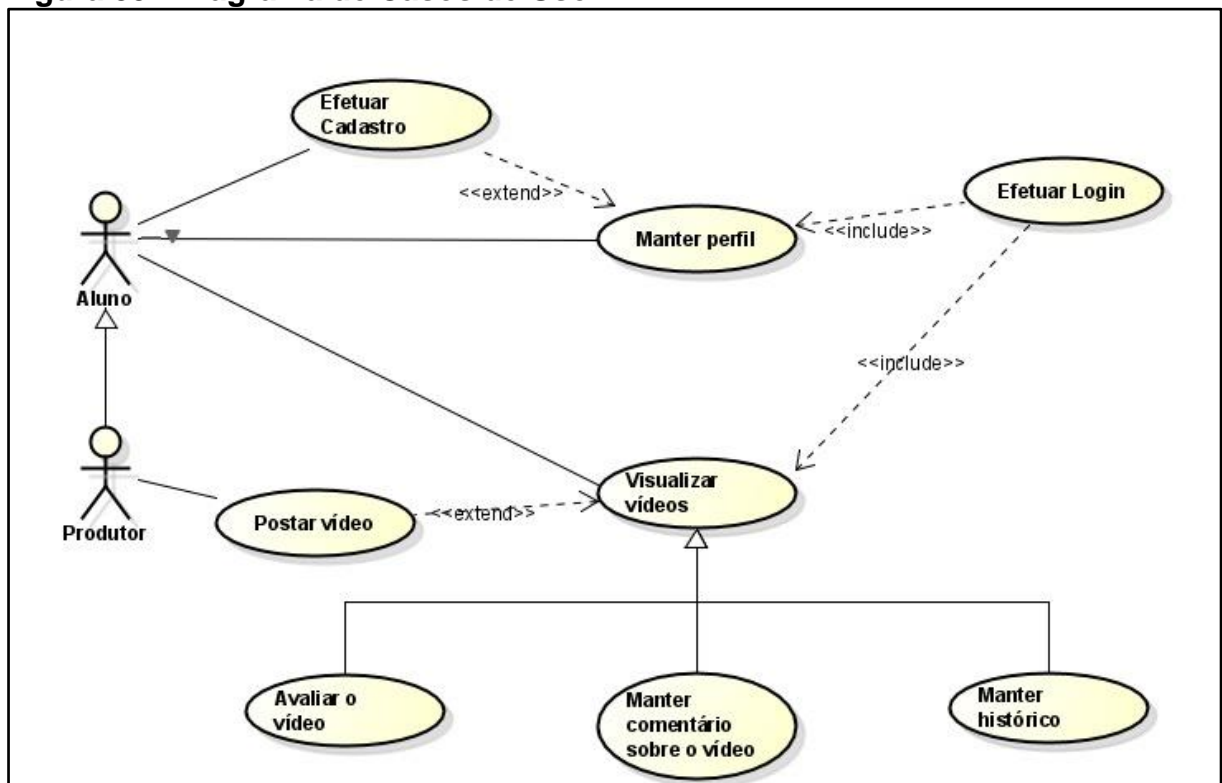
3 DESENVOLVIMENTO

Nesse capítulo será apresentado o desenvolvimento da plataforma GetJob, por meio dos diagramas criados no estudo UML, elaboração do banco de dados e as tecnologias utilizadas no desenvolvimento do projeto.

3.1 Diagrama de Casos de Uso

Por meio da Figura 39, é possível ver, através do caso de uso, as ações e utilidades que o usuário poderá realizar no site. Demonstrando o meio em que as funcionalidades do site podem fazer e seus requisitos para funcionar:

Figura 39 - Diagrama de Casos de Uso

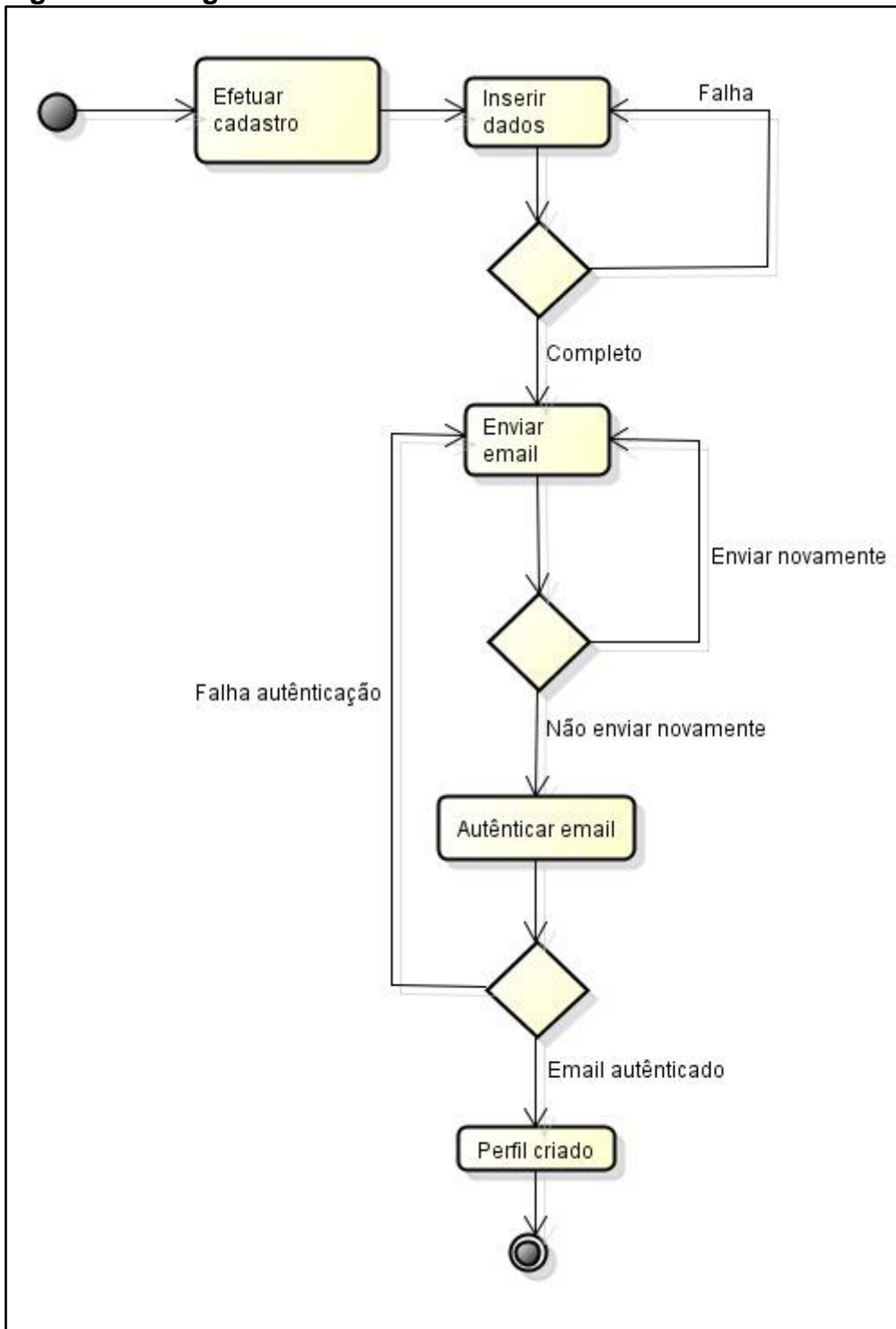


Fonte: Autoria própria, 2022.

3.2 Diagrama de Atividades

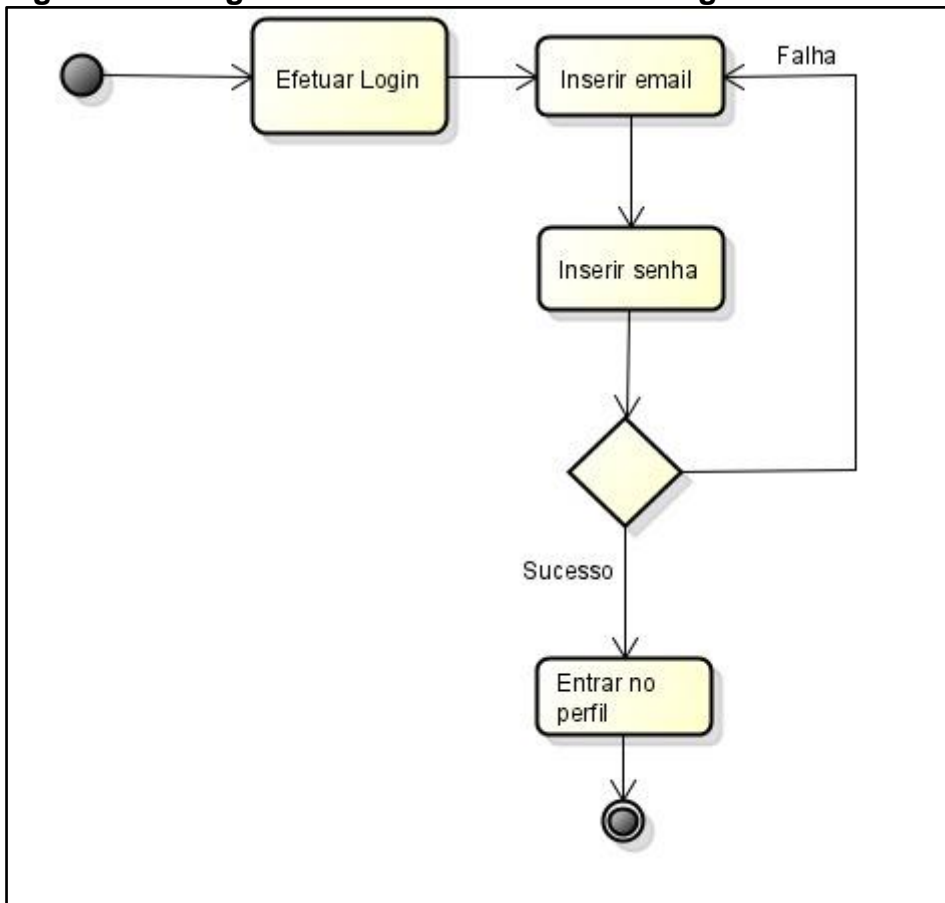
A Figura 40, neste caso, representa o diagrama das atividades realizadas quando um usuário tenta se registrar no site, em caso de um cadastro de conta normal, terá o preenchimento dos dados, a autenticação do e-mail e então, por fim, a criação do perfil:

Figura 40 - Diagrama de Atividade: Efetuar Cadastro



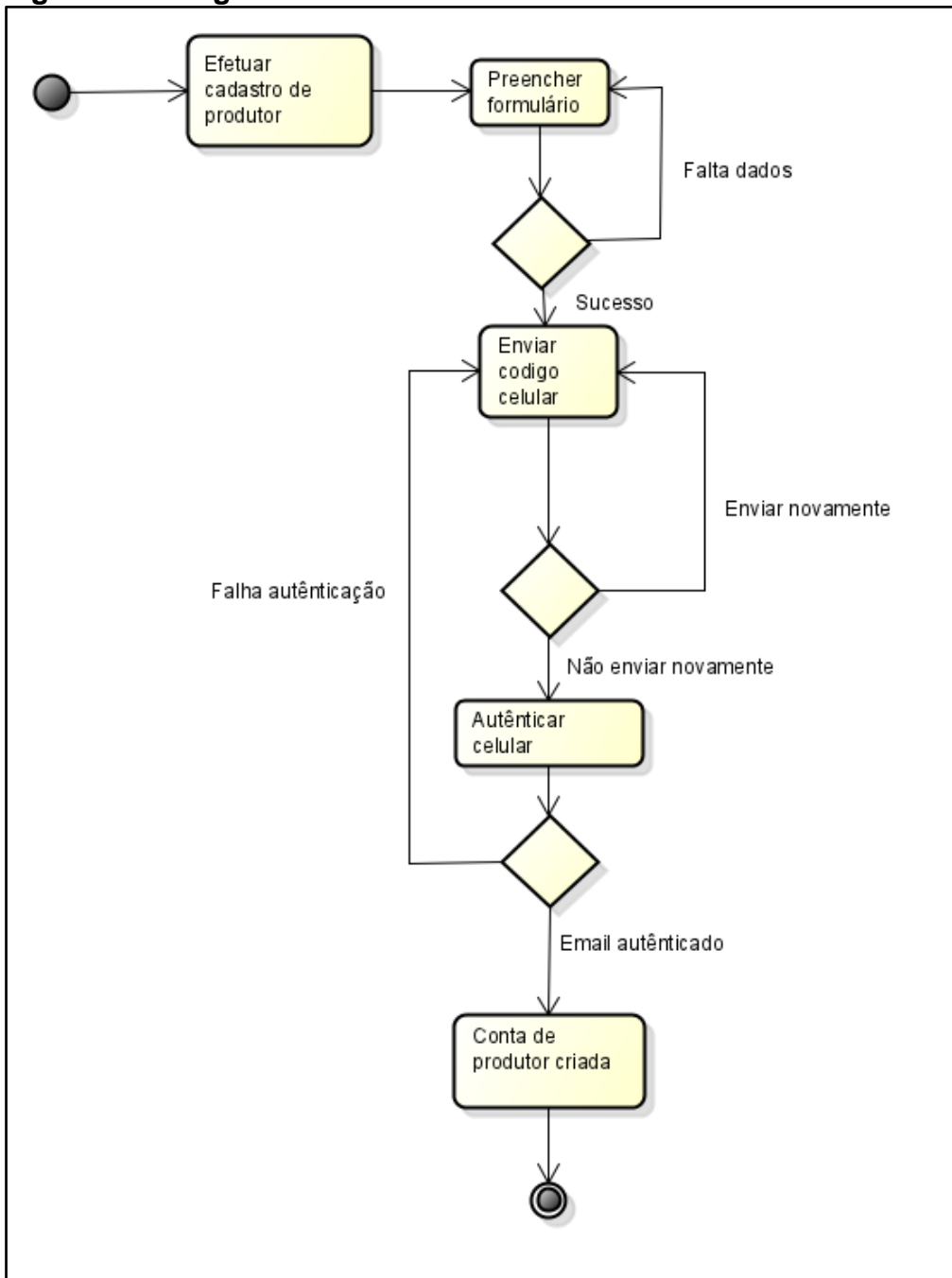
Fonte: Autoria própria, 2022.

A Figura 41, representa a atividade de login, sendo a sequência das atividades seguidas quando uma tentativa de login é realizada, por meio da inserção dos dados e a validação do e-mail e senha para verificar o usuário:

Figura 41 - Diagrama de Atividade: Efetuar Login

Fonte: Autoria própria, 2022.

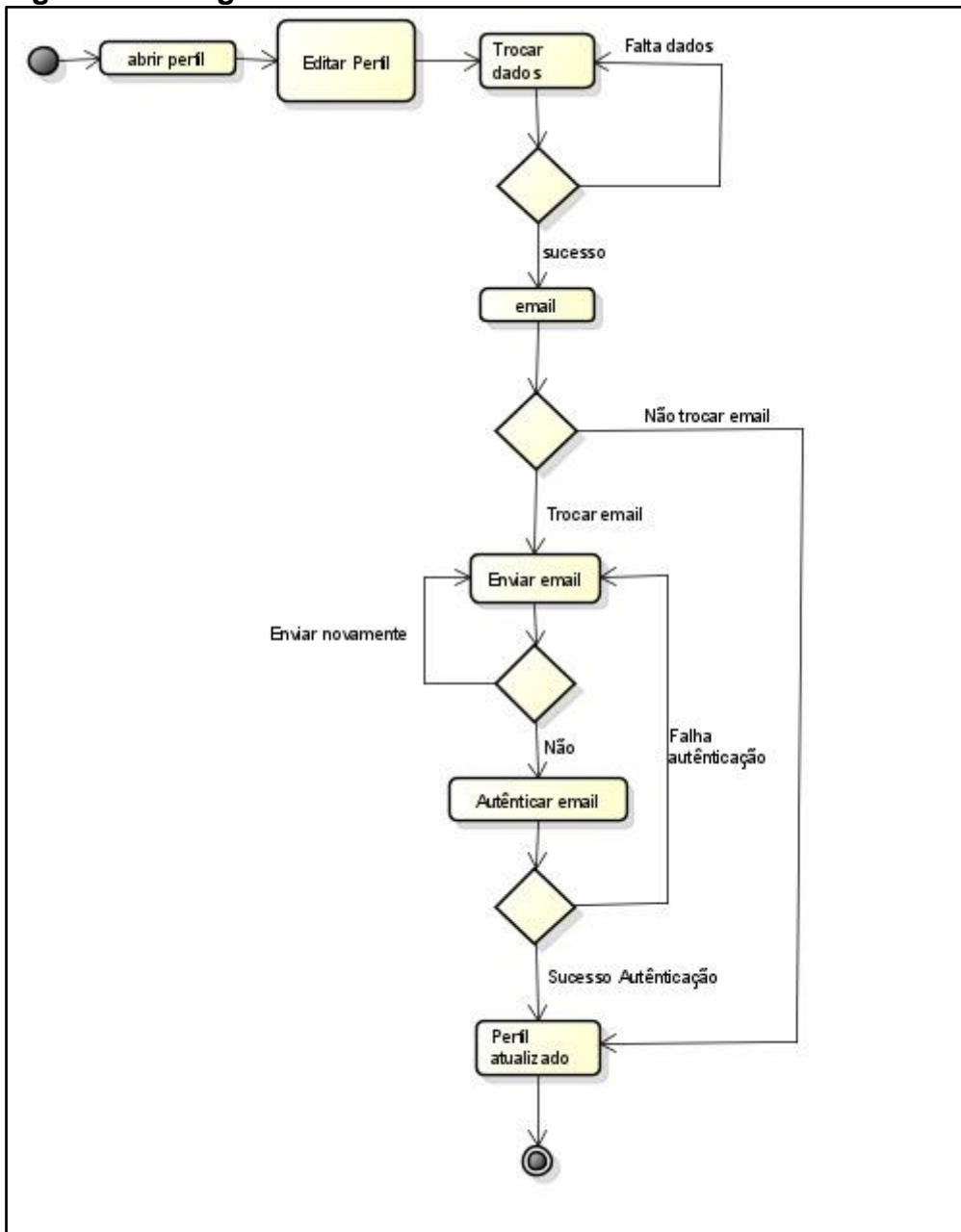
Na Figura 42, é exibido o fluxo quando se está fazendo a criação de uma conta de produtor, sendo que, nesta sequência, o usuário precisa já ter conta no site, e então, ela poderá tornar sua conta como a de um produtor ao selecionar essa opção e preencher o formulário. Quando completamente preenchido, ocorrerá a autenticação do e-mail e, pôr fim, a conta se torna uma conta de produtor:

Figura 42 - Diagrama de Atividade: Cadastro de Conta do Produtor

Fonte: Autoria própria, 2022.

Na Figura 43, é representado o fluxo de atividades no momento da troca de informações da conta:

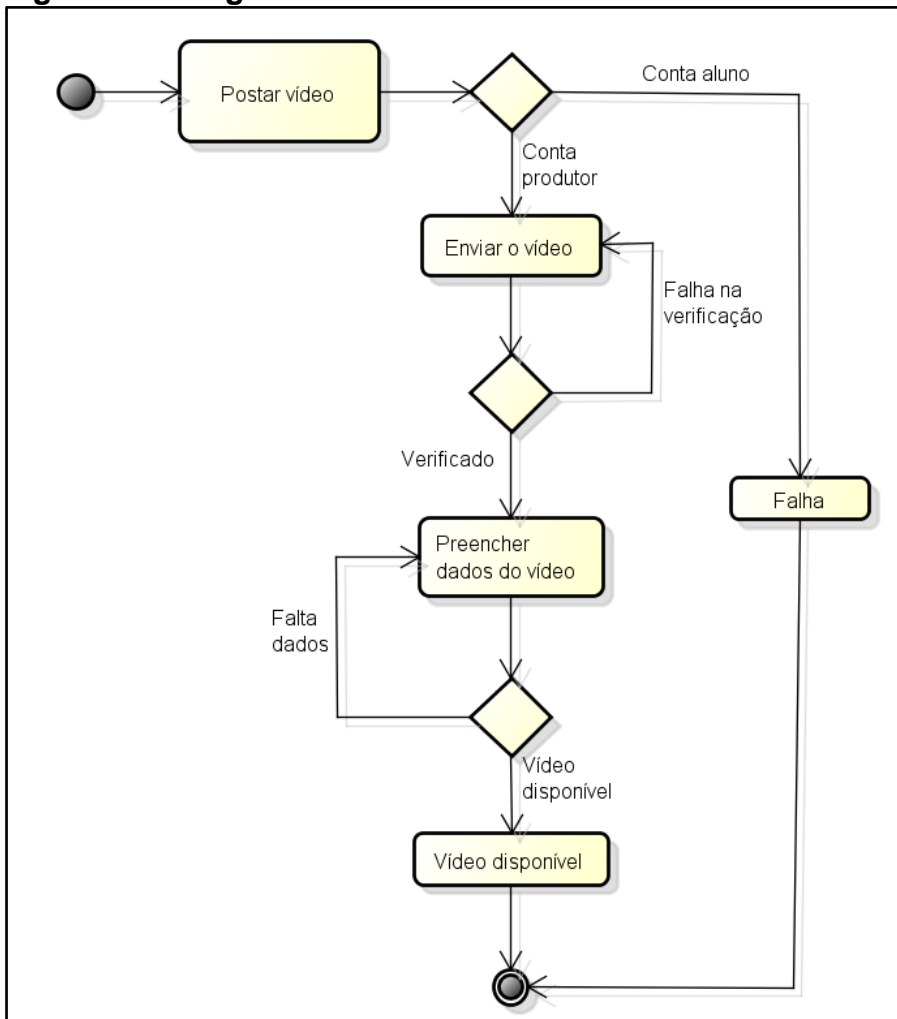
Figura 43 - Diagrama de Atividade: Editar Perfil



Fonte: Autoria própria, 2022.

A Figura 44, mostra como é o processo do envio de vídeo para o site e os requisitos para que ele apareça aos outros usuários:

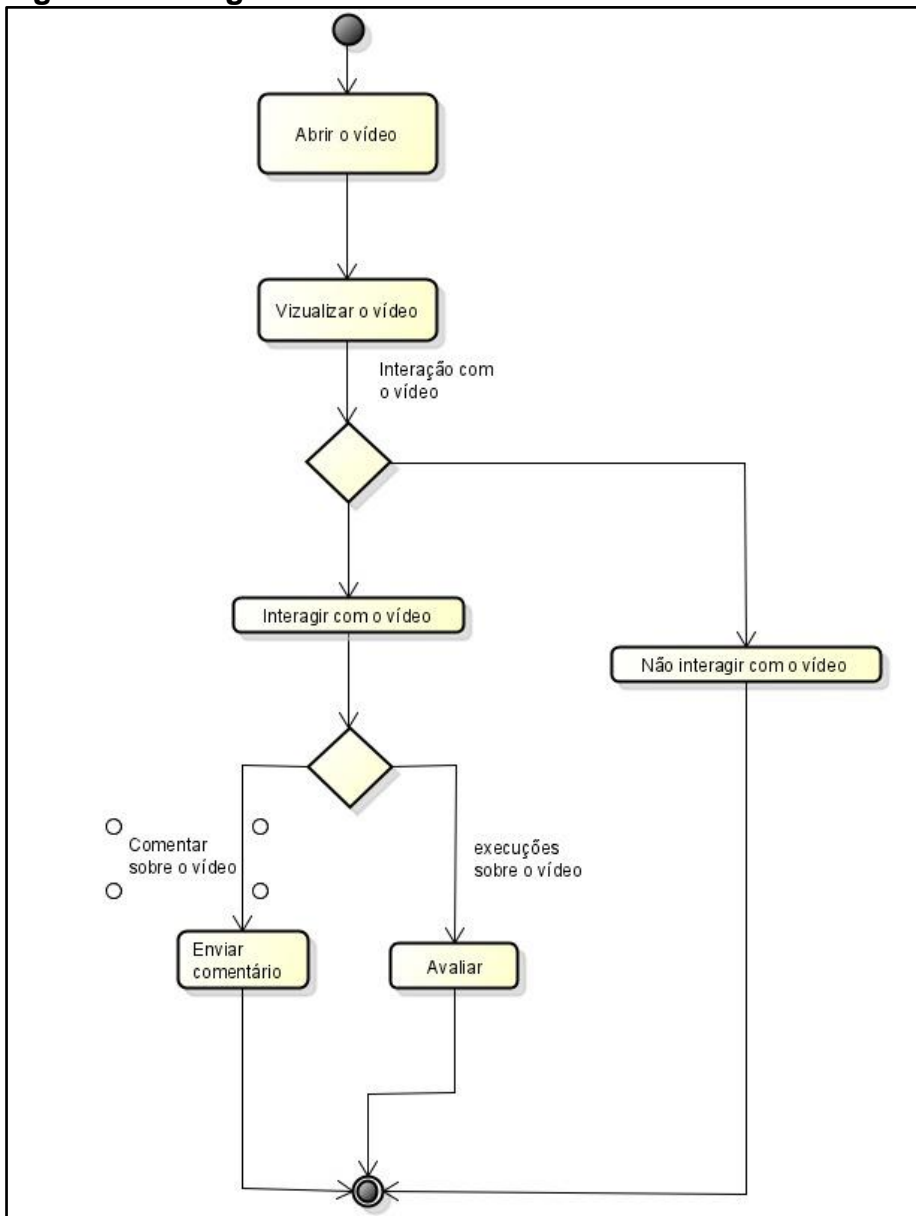
Figura 44 - Diagrama de Atividade: Publicar Vídeo



Fonte: Autoria própria, 2022.

Na Figura 45, é demonstrado o curso de atividades que um usuário pode ter na hora de acessar e ver um vídeo:

Figura 45 - Diagrama de Atividade: Atividades do Vídeo



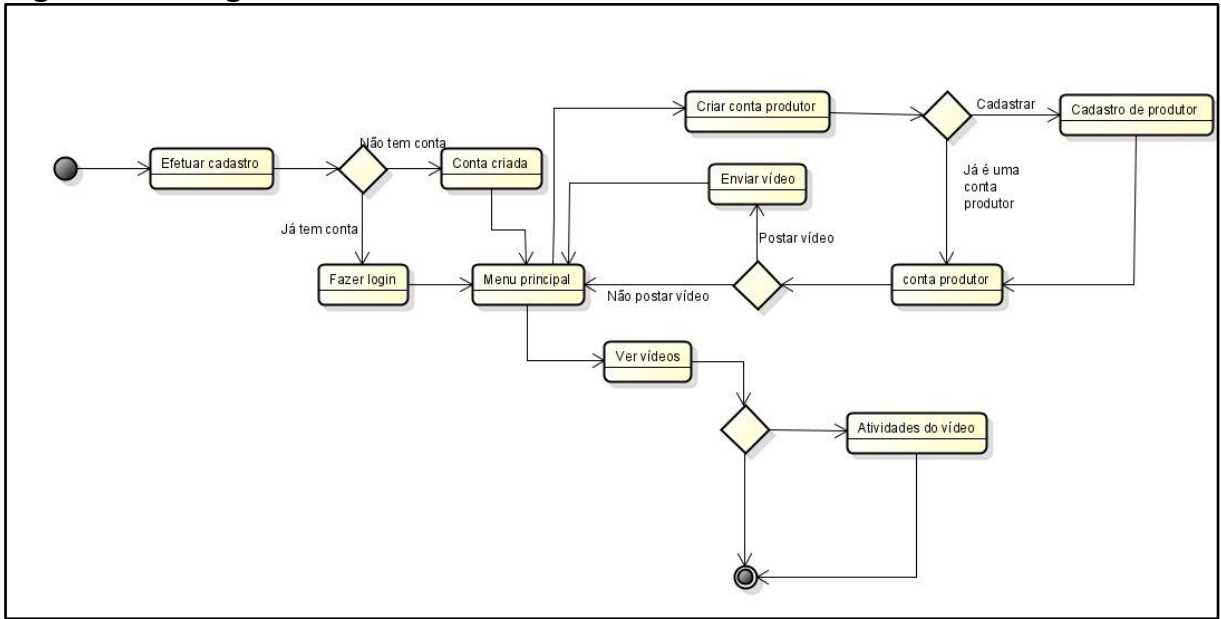
Fonte: Autoria própria, 2022.

Representando as ações que um usuário poderá realizar na página de vídeo, a Figura 45 demonstra todas as funções que podem ser realizadas na página de vídeo.

3.3 Diagrama de Estado

É apresentado o diagrama de estado, Figura 46, diagrama o qual demonstra o processo geral de funções que o site executa:

Figura 46 - Diagrama de Estado

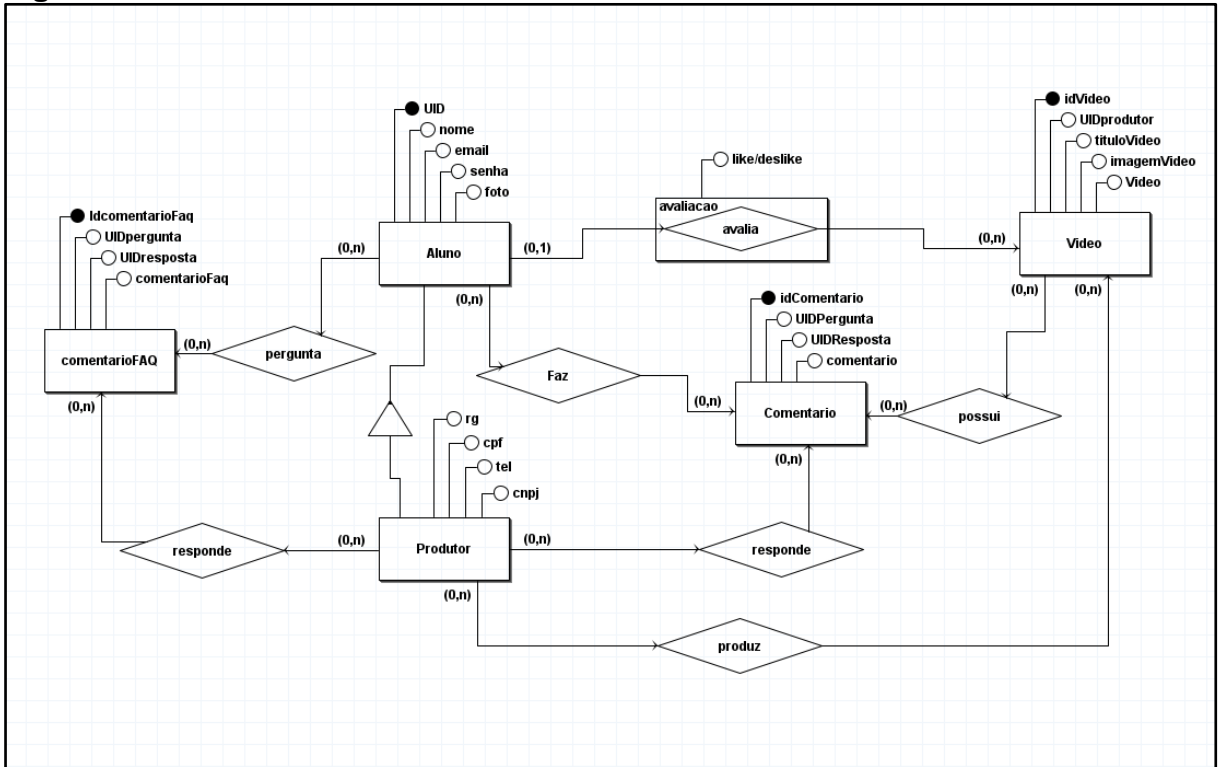


Fonte: Autoria Própria, 2022.

3.4 DER

Os diagramas de entidade relacionamento (DER), são a composição visual para que se possa ter a noção das funções e eventos que serão executados no site quando se diz em questão ao banco de dados.

Figura 47 - DER



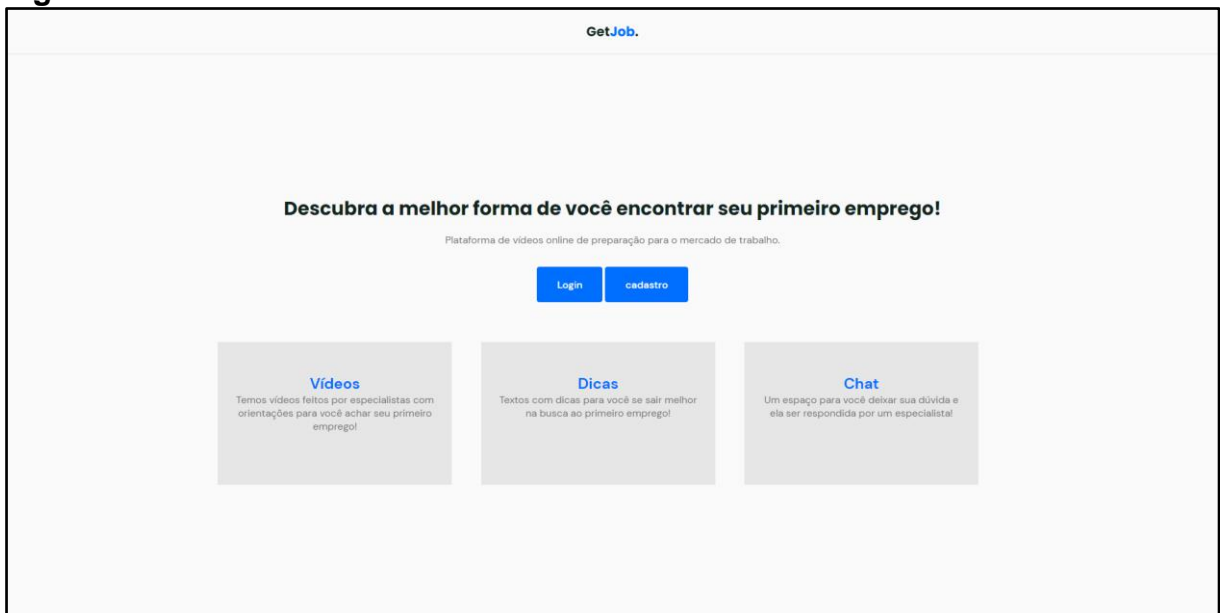
Fonte: Autoria Própria, 2022.

Representando de forma visual como o site funcionara em relação ao banco, a Figura 47 retrata quais são os eventos que um usuário pode realizar no site e as funções que serão geradas a partir desse evento. Como quando o usuário fizer um comentário no vídeo; a partir dessa ação, é gerada uma entidade “comentário” no banco, para que vire uma parte daquela página.

3.5 Telas

Sendo a primeira página do site, a Figura 48 é a primeira coisa que o usuário tem acesso, disponibilizando o link para o cadastro e login.

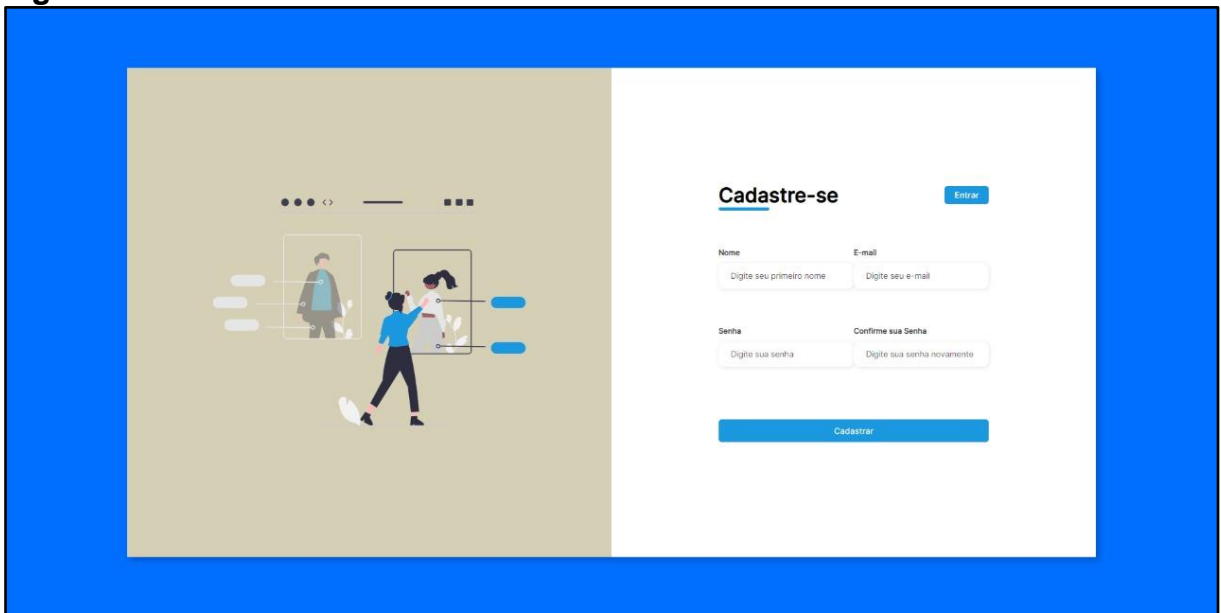
Figura 48 – Tela Inicial: Home



Fonte: Autoria Própria, 2022.

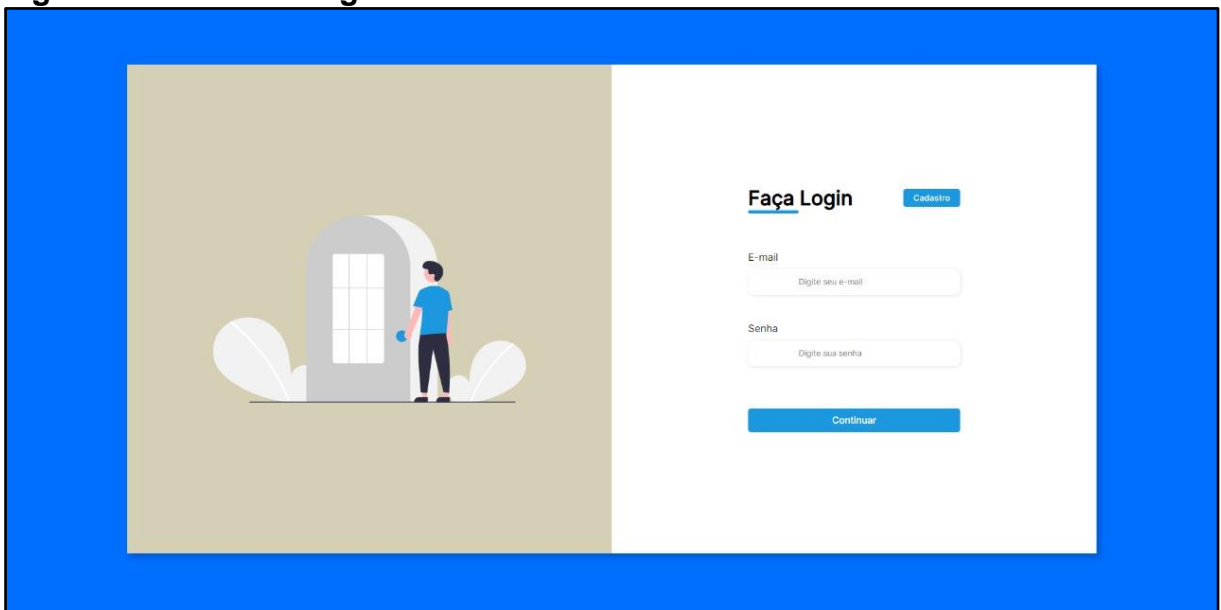
Sequências da página inicial, as Figuras 49 e 50 são as páginas para que o usuário possa ter acesso ao site, sendo necessário que ele faça seu cadastro, caso já o tenha feito, então que faça seu login no site.

Figura 49 – Tela de Cadastro



Fonte: Autoria Própria, 2022.

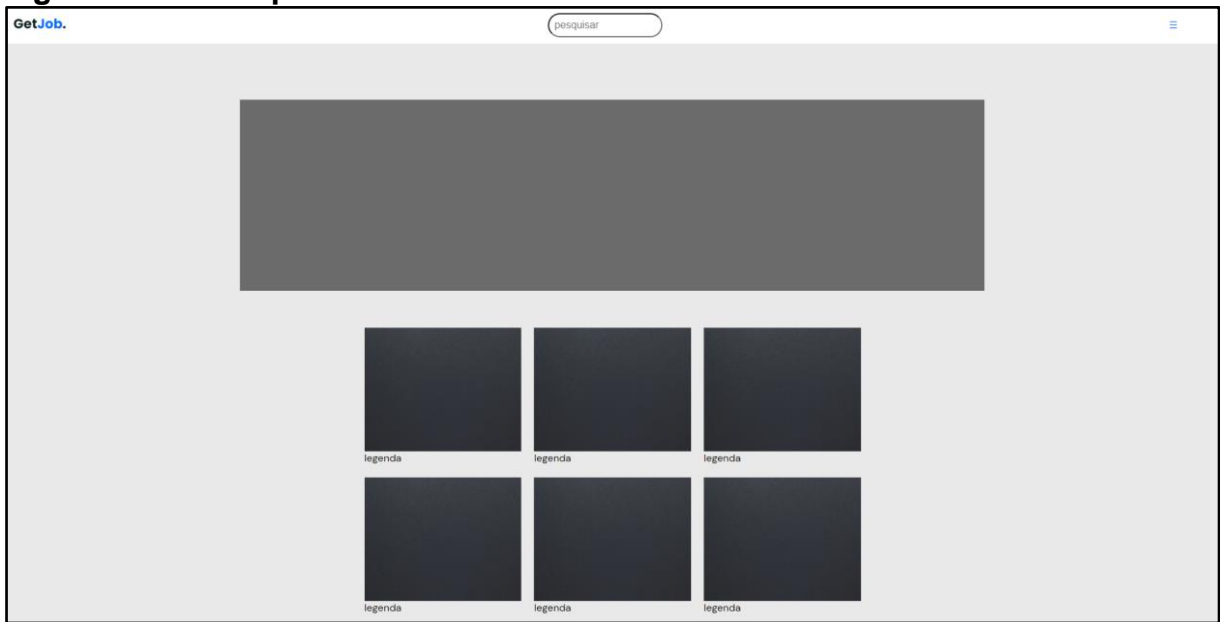
Figura 50 – Tela de Login



Fonte: Autoria Própria, 2022.

Logo após ter sua conta criada, o usuário, como aluno, terá acesso ao que é a página principal do vídeo, a página de vídeos, Figura 51. Nesta página, será apresentada uma variedade de vídeos para que o usuário possa assistir ou pesquisar por temas específicos; também podendo acessar quaisquer outras páginas disponibilizadas na plataforma. Os retângulos pretos representam, respectivamente, onde ficariam as *thumbnails*, a imagem utilizada como capa do vídeo, para que o usuário possa selecionar ao que assistir. O retângulo cinza é um recinto para a exibição de anúncios no site.

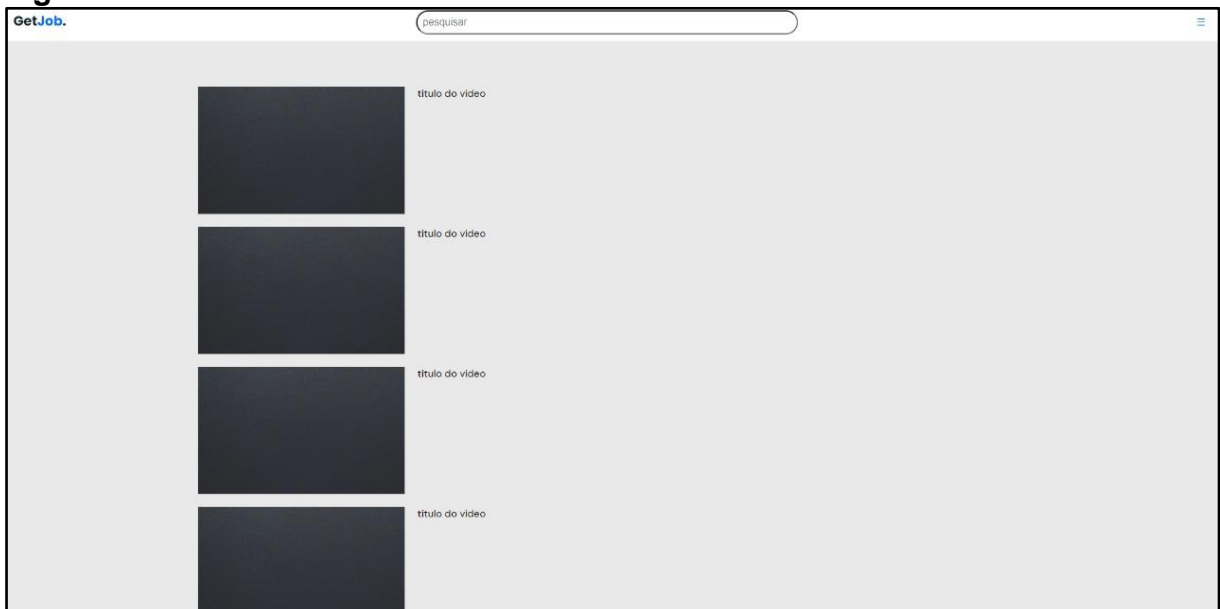
Figura 51 – Principal Tela de Vídeos



Fonte: Autoria Própria, 2022.

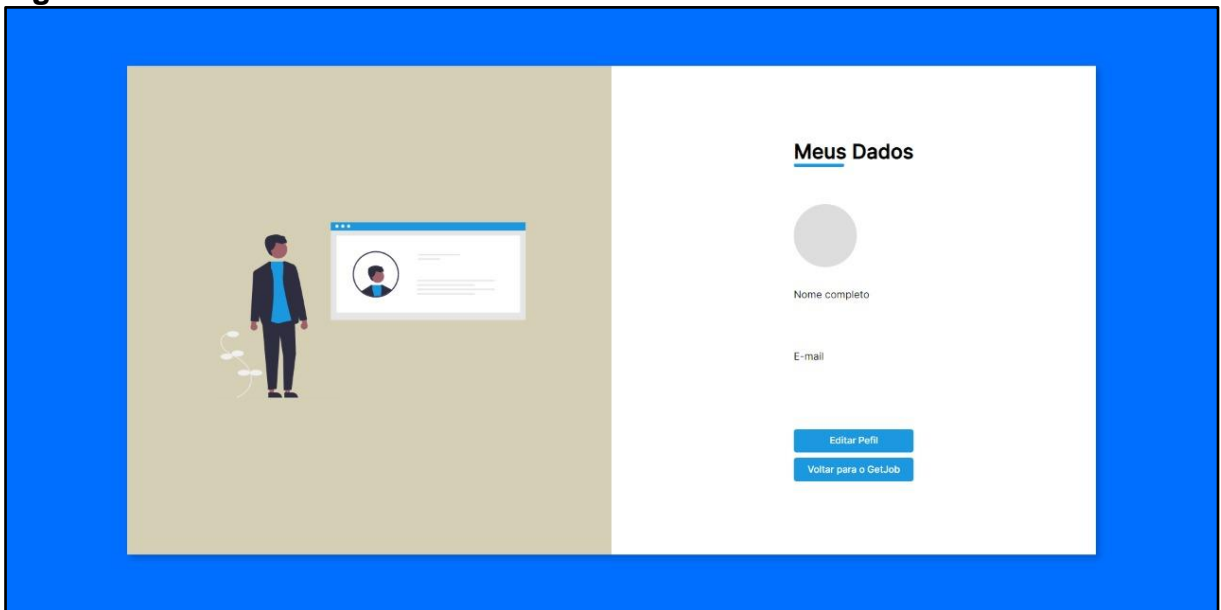
Sendo uma extensão da página de vídeos, a tela de busca de vídeos, representada pela Figura 52, demonstra como será para o usuário ao buscar por um tipo específico de vídeo. Os retângulos pretos são um exemplo de como seria a filtragem entre os tipos de vídeos que a pessoa está procurando.

Figura 52 - Tela de Busca

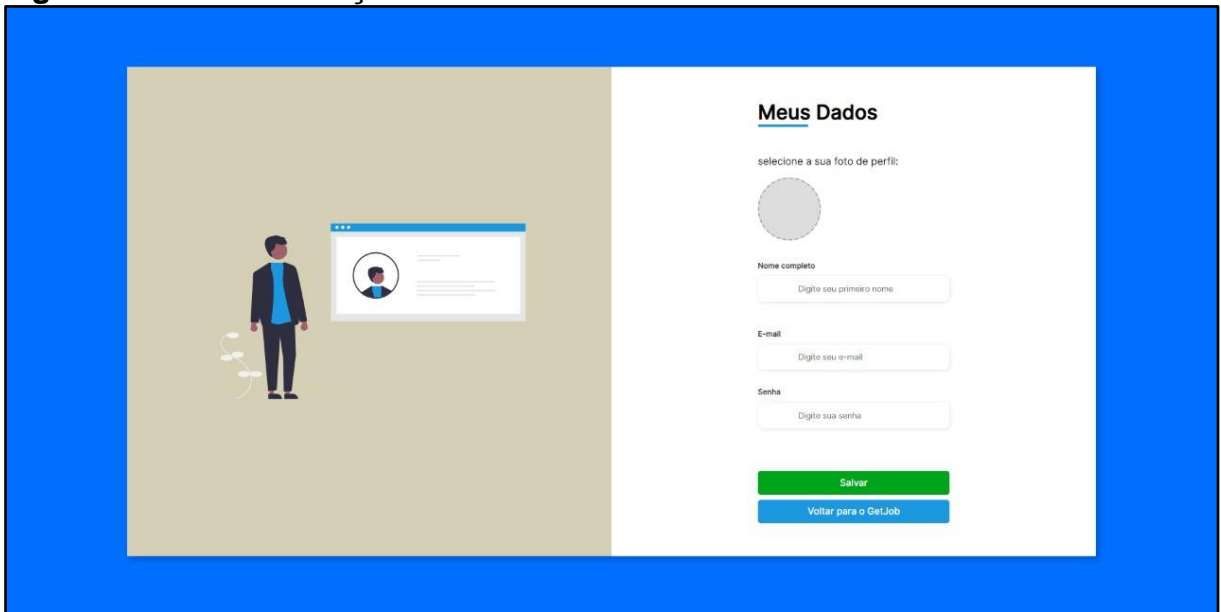


Fonte: Autoria Própria, 2022.

É possível que o usuário possa checar e editar suas informações a qualquer momento; sendo a Figura 53, a página para a checagem das informações de perfil e, a Figura 54, a página para alterá-las.

Figura 53 - Tela de Perfil

Fonte: Autoria Própria, 2022.

Figura 54 – Tela de Edição de Perfil

Fonte: Autoria Própria, 2022.

Para qualquer aluno que tenha interesse em ajudar ou compartilhar seu próprio conhecimento, é possível que ele faça seu cadastro como produtor na página de cadastro especial, exibida na Figura 55, onde ele terá que inserir mais alguns dados, como forma de controle sobre quem serão os produtores.

Figura 55 - Tela de Cadastro do Produtor

Cadastro de Produtor

Para enviar um vídeo faça o cadastro especial!

Nome completo Digite seu nome completo

RG Digite seu RG

CNPJ Digite sua CNPJ

CPF Digite seu CPF

Telefone Digite seu número de telefone

Confirme sua Senha Digite sua senha novamente

[Cadastro especial!](#)

[Voltar para o GetJob](#)

Fonte: Autoria própria, 2022.

Tendo feito seu cadastro como produtor, o usuário poderá compartilhar seus próprios vídeos no site, preenchendo os dados do vídeo como mostrado na Figura 56.

Figura 56 – Tela de Envio de Vídeo

Enviar Vídeo

seleccione a capa do seu vídeo

Choose an image

seleccione seu vídeo

Escolher ficheiro Nenhum ficheiro selecionado

Titulo do vídeo

Digite o título

Descrição do vídeo

Digite a Descrição

[Enviar vídeo](#)

Fonte: Autoria Própria, 2022.

A Figura 57 apresenta como é a tela para visualização dos vídeos escolhidos pelo usuário, que então, ao selecionar um vídeo de seu interesse na principal página de vídeos, demonstrada pela Figura 51, ou então pela página de procura de vídeos, na Figura 52, fará o usuário ser redirecionado para tela de vídeo. Sendo que nesta página, é possível visualizar o vídeo, avaliar ele ou realizar um comentário.

Figura 57 - Tela de Vídeo

Fonte: Autoria Própria, 2022.

4 CONCLUSÃO

Esse trabalho pretendeu criar uma plataforma de vídeos educativos com o intuito de compartilhar conteúdo para auxiliar jovens a ingressar no mercado de trabalho, dado que, segundo a Organização Internacional do Trabalho (2021), os jovens de 15 a 24 anos se caracterizaram como grupo mais atingido pelo desemprego, e pela falta de acesso a um conteúdo educativo que ensine o que o atual mercado de trabalho procura, esse problema tende a piorar com os anos.

Cumprindo o objetivo, foi desenvolvido o GetJob: plataforma de vídeos online, que oferece aos usuários acesso a vídeos e a uma comunidade em que os membros podem se ajudar. Com o foco em ser um ambiente amigável, onde é possível comentar e trocar experiências, ensinar de maneira coesa através de um conteúdo pensado para fácil compreensão. Onde o foco é ser uma plataforma acessível, sendo simples e intuitiva.

A partir do estudo de UML, HTML, CSS, JavaScript e Firebase. Com os diagramas de caso de uso, atividade e estado, tornou-se possível um planejamento robusto, visando a orientação para criação. Foi desenvolvida a plataforma onde o usuário logado tem acesso a todos os vídeos disponíveis, podendo filtrar através da barra de pesquisa, sendo possível adicionar comentários aos vídeos, permitindo que sejam compartilhadas opiniões, dúvidas e/ou críticas construtivas. O usuário pode escolher entre ser um membro comum da plataforma ou se tornar um produtor de vídeos, através do cadastro especial, onde são inseridos dados, entre eles, certificações digitalizadas, para comprovar algum conhecimento. Será averiguado se o documento é confiável, para que o usuário receba permissão para publicar vídeos, caso seja um certificado válido, o usuário é autorizado e poderá criar e compartilhar seu conteúdo. Ao despertar interesse de terceiros, GetJob pode se tornar uma empresa mais elaborada, trabalhando com monetização e gerando lucros para usuários e criadores.

REFERÊNCIAS

ALEX SCHEDT, Felipe. **Fundamentos de CSS: criando Design para Sistemas Web**. Foz do Iguaçu: Outbox Interativa, 2015. 120 p.

BEZERRA, Eduardo. **Princípios de análise e Projeto de sistemas com UML**. 2. ed. Rio de Janeiro: Elsevier Brasil, 2006. 392 p.

BLAHA, Michael; RUMBAUGH, James. **Modelagem e Projetos Baseados em Objetos com UML 2**. 2. ed. Rio de Janeiro: Elsevier Brasil, 2006. 520 p.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário**. 2. ed. Rio de Janeiro: Elsevier Brasil, 2006. 474 p. Tradução de: Fábio Freitas da Silva.

CARDOSO, Virínia Mara; CARDOSO, Giselle Cristina. **Sistema de Banco de Dados**. São Paulo: Saraiva Uni, 2012. 144 p.

Crise do emprego juvenil: desarmando a bomba-relógio ativada pela pandemia. Disponível em: <https://www.ilo.org/brasilia/noticias/WCMS_816644/lang-pt/index.htm>. Acesso em: 6 dez. 2022.

DATE, Christopher J. **Introdução a sistemas de banco de dados**. Rio de Janeiro: Elsevier, 2003. 896 p. Tradução de: Daniel Vieira.

DEITEL, Paul; DEITEL, Harvey. **Java®: Como Programar**. 6. ed. São Paulo: Pearson Universidades, 2005. 968 p. Tradução de: Edson Furmankiewicz.

Documentação do Firebase. Disponível em: <<https://firebase.google.com/docs?authuser=3&hl=pt>>. Acesso em: 6 dez. 2022.

EIS, Diego. **HTML5 e CSS3 com farinha e pimenta: Desenvolvimento client-side inteligente**. São Paulo, 2012. 167p.

ELMASRI, Ramez; NAVATHE, Shamkant B. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson Addison Wesley, 2005. 808 p.

FLANAGAN, David. **JavaScript - O Guia Definitivo**. 6. ed. Porto Alegre: Bookman, 2011. 1080 p.

FLATSCHART, Fábio. **HTML 5 - Embarque imediato**. Rio de Janeiro: Brasport, 2011. 223 p.

FOWLER, Martin. **UML Essencial: um breve guia para a linguagem-padrão de modelagem para objetos**. 3. ed. Porto Alegre: Bookman Editora, 2004. 162 p.

GUEDES, Gilleanes T. A. **UML 2 - Uma Abordagem Prática**. 3. ed. São Paulo: Novatec Editora, 2018. 496 p.

GUEDES, Gilleanes T. A. **UML 2 - Uma Abordagem Prática**. 2. ed. São Paulo: Novatec Editora, 2011. 488 p.

LARMAN, Craig. **Utilizando UML e Padrões**. 3. ed. Rio de Janeiro: Bookman, 1997. 492 p.

LÓSCIO, Bernadette Farias; OLIVEIRA, Hélio Rodrigues de; PONTES, Jonas César de Sousa. **NoSQL no desenvolvimento de aplicações Web colaborativas**. 2011. 35 f. Tese (Doutorado) - Curso de Banco de Dados, Simpósio Brasileiro de Sistemas Colaborativos, Rio de Janeiro, 2011.

MACHADO, Kheronn Khennedy; OLIVEIRA, Hélio Rodrigues de; PONTES, Jonas César de Sousa. **Angular 11 e Firebase: construindo uma aplicação integrada com a plataforma do google**. São Paulo: Casa do Código, 2021. 193 p.

MANARA MILETTO, Evandro; de Castro Bertagnolli, Silvia. **Desenvolvimento de software: introdução ao desenvolvimento web com HTML, C55, JavaScripte. PHP.** - Porto Alegre: Bookman, 2014. 265 p.

PANIZ, David. **NoSQL: como armazenar os dados de uma aplicação moderna**. São Paulo: Casa do Código, 2016. 218 p.

PRESCOTT, Preston. **Programação em JavaScript**. Rio de Janeiro: Babelcube Inc., 2016. 73 p. Tradução de: Fernando Souza.

PRESSMAN, Roger S.; MAXIM, Bruce R.; ARAKAKI, Julio; ARAKAKI, Reginaldo; ANDRADE, Renato Manzan de. **Engenharia de software: uma abordagem profissional**. 9. ed. Porto Alegre: Amgh, 2021. 704 p. Tradução de: Ferraretto e Costa Ltda.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de gerenciamento de banco de dados**. 3. ed. Porto Alegre: Amgh, 2011. 905 p. Tradução de: Célia Taniwake.

SADALAGE, Pramod J.; FOWLER, Martin. **NoSQL Essencial: um guia conciso para o mundo emergente da persistência poliglota**. São Paulo: Novatec Editora, 2013. 216 p.

SILVA, Maurício Samy. **Construindo Sites com CSS e (X)HTML: sites controlados por folhas de estilo em cascata**. São Paulo: Novatec, 2008. 439 p.

SILVA, Maurício Samy. **Fundamentos de HTML5 e CSS**. São Paulo: Novatec, 2015. 304 p.

SILVA, Maurício Samy. **JavaScript - Guia do Programador: guia completo das funcionalidades de linguagem javascript**. São Paulo: Novatec Editora, 2010. 608 p.

STADZISZ, Paulo César. **Projeto de Software Usando UML**. Disponível em: <<http://www.etelg.com.br/paginaete/downloads/informatica/apostila2uml.pdf>>. Acesso em: 12 out. 2022.