

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

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

Ana Karine Bomfim Cardoso

Emily Leme Silva

Gustavo Cardoso Barros

Ítallo da Silva Santos

Kawê Cavalcante Ribeiro

Search Devs

**São Paulo
2022**

Ana Karine Bomfim Cardoso

Emily Leme Silva

Gustavo Cardoso Barros

Ítallo da Silva Santos

Kawê Cavalcante Ribeiro

Search Devs

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 pela professora Vilma Cardoso e pelos professores Ediney Ciasi Barreto e Wagner Lucca, como requisito final para obtenção do título de Técnico em Desenvolvimento de Sistemas.

**São Paulo
2022**

DEDICATÓRIA

Dedicamos esse trabalho a todos amigos e familiares por todo o incentivo e ajuda para que fosse possível a conclusão desse TCC.

AGRADECIMENTOS

Agradecemos aos professores orientadores que nos acompanharam durante o processo de desenvolvimento desse projeto de conclusão de curso, oferecendo todo o apoio possível, em especial a professora Vilma Cardoso dos Santos.

Agradecemos também a todos que participaram das pesquisas e codificação, pela disposição e colaboração.

EPÍGRAFE

“Se você realmente acredita naquilo que faz e está disposto a trabalhar duro, então você provavelmente será bem-sucedido.”

PIERRE OMIDYAR, FUNDADOR DA EBAY

RESUMO

A demanda por tecnologia aumentou muito nos anos de 2020 a 2022, no futuro será muito comum trabalhos independentes desenvolvidos por meios de plataformas digitais. Com isso surgiu a necessidade de desenvolver algo que aproximasse contratado e contratante, então foi criado o projeto Search Devs, uma plataforma digital que tem como objetivo diminuir o déficit de empregos na área da tecnologia e gerar relações profissionais entre desenvolvedores e empresa. Esse trabalho foi feito a partir de levantamento de dados através de pesquisas bibliográficas e análise documental. Ao obter os resultados das pesquisas, foram propostas melhorias para o meio de contratação.

Palavras-chave: trabalhos independentes, contratação, desenvolvedores

ABSTRACT

The demand for technology increased a lot in the years 2020 to 2022, in the future independent work developed through digital platforms will be very common. With that, the need arose to develop something that would bring contractor and contractor closer together, so the search devs project was created, a digital platform that aims to reduce the deficit of jobs in the technology area and generate professional relationships between developers and the company. This work was done from data collection through bibliographic research and document analysis. Upon obtaining the results of the surveys, improvements were proposed for the means of hiring.

Keywords: independent work, hiring, developers

LISTA DE ILUSTRAÇÕES

Figura 1 - Estrutura Básica HTML5	14
Figura 2 - Código da Tela de Cadastro	16
Figura 3 - Tela de Cadastro	17
Figura 4 - Estrutura Básica de CSS	17
Figura 5 - Tag de Estilo	18
Figura 6 - Código de Estilo de Cadastro	18
Figura 7 - Cadastro com CSS	19
Figura 8 - tag Link Bootstrap	20
Figura 9 - tag Script Bootstrap	20
Figura 10 - Bootstrap na Prática	21
Figura 11 - Tela com Bootstrap	22
Figura 12 - Script Javascript	23
Figura 13 - Validação com Javascript	23
Figura 14 - Alerta de Javascript	24
Figura 15 - Exemplo de JQuery	25
Figura 16 - Banco de Dados MySQL	26
Figura 17 - Cadastro no Banco de Dados	27
Figura 18 - Resultado no Banco de Dados	27
Figura 19 - Formulário Login PHP	27
Figura 20 - Formulário Cadastro PHP	27
Figura 21 - Cadastro PHP	28
Figura 22 - Login PHP	28
Figura 23 - Classe de Modelagem	29
Figura 24 - Exemplo de Caso de Uso	32
Figura 25 - Exemplo de Diagrama de Classe	33
Figura 26 - Exemplo de Diagrama de Sequência	34
Figura 27 - Exemplo de Diagrama de Atividade	34
Figura 28 - Diagrama de Caso de Uso do Search Devs	39
Figura 29 - Diagrama de classes do Projeto Search Devs	40
Figura 30 - Diagrama de Atividades do Cadastro	41
Figura 31 - Diagrama de Atividades Login	42
Figura 32 - Diagrama de Atividades Gerais 1	43
Figura 33 - Diagrama de Atividades Gerais 2	44
Figura 34 - Diagrama de Sequência Cadastro e Login do Contratante	44
Figura 35 - Diagrama de Sequência de Alteração de Imagem do Contratante	45
Figura 36 - Diagrama de Sequência de Pesquisa de Perfis do Contratante	45
Figura 37 - Diagrama de Sequência da Criação de Projeto do Contratante	45
Figura 38 - Diagrama de Sequência Visualizar Desenvolvedor Contratante	46
Figura 39 - Diagrama de Sequência Visualizar Projeto do Contratante	46
Figura 40 - Diagrama de Sequência do Cadastro e Login do desenvolvedor	47
Figura 41 - Diagrama de Sequência do Cadastro e Login do Desenvolvedor 2	47
Figura 42 - Diagrama de Sequência Alteração de Imagem Desenvolvedor	48
Figura 43 - Diagrama de Sequência de Pesquisa Perfil Desenvolvedor	48
Figura 44 - Diagrama de Sequência Visualizar Propostas Desenvolvedor	49
Figura 45 - DER do Search Devs	49
Figura 46 - Home	50
Figura 47 - Tela de escolha de Cadastro	51
Figura 48 - Tela de Cadastro Empreendedor	51
Figura 49 - Tela de Cadastro Desenvolvedor	52
53	

Figura 50 - Tela de Cadastro Área	53
Figura 51 - Tela de Cadastro Desenvolvedor Habilidades	54
Figura 52 - Perfil do Desenvolvedor	55
Figura 53 - Perfil Contratante	55
Figura 54 - Perfil Contratante 2	56
Figura 55 - Tela de Login	57
Figura 56 - Pesquisar Perfis	57
Figura 57 - Projetos em Andamento	58
Figura 58 - Criação de Projeto	59
Figura 59 - Desenvolvedores recomendados	59
Figura 60 - Novas Propostas	60

LISTA DE TABELAS

Tabela 1 - Tabela dos Requisitos Funcionais do Search Devs	35
Tabela 2 - Tabela dos Requisitos Não Funcionais do Search Devs	37
Tabela 3 - Tabela das Regras de Negócio do Search Devs	37

LISTA DE ABREVIATURAS E SIGLAS

Application Programming Interface (API)

Cascading Style Sheet (CSS)

HyperText Markup Language 5 (HTML5)

Hypertext Preprocessor (PHP)

Internacional Business Machines (IBM)

Sistemas Gerenciadores de Banco de Dados (SGBD)

Structured Query Language (SQL)

Unified Modeling Language (UML)

Sumário

1 2	INTRODUÇÃO	13
	REFERENCIAL TEÓRICO	14
2.1	HTML5	14
2.2	CSS	17
2.3	Bootstrap	19
2.4	JavaScript	22
2.5	JQuery	24
2.6	Banco de Dados	25
2.6.1	MySQL	26
2.7	PHP	27
2.8	UML	30
2.8.1	Requisitos de Software	31
2.8.2	Diagrama de Caso de Uso	32
2.8.3	Diagrama de Classes	33
2.8.4	Diagrama de Sequência	33
3 2.8.5	Diagrama de Atividade	34
	DESENVOLVIMENTO	35
3.1	Levantamento de Requisitos do Search Devs	35
3.2	Diagrama de Caso de Uso do Search Devs	38
3.3	Diagrama de classes do Search Devs	39
3.4	Diagrama de atividades do Search Devs	40
3.5	Diagrama de Sequência do Search Devs	44
3.6	DER	50
43.7	Telas Search Devs	50
	CONCLUSÃO	61

1 INTRODUÇÃO

É de conhecimento geral que a tecnologia se disseminou pelo mundo nos últimos anos, tendo grande importância no período pandêmico, segundo Caetano (2021), isso levou ao crescimento na demanda do setor da informática e conseqüentemente gerou uma dificuldade em selecionar profissionais qualificados.

De acordo com Araújo (2022), em uma pesquisa realizada por Harvard Business Review, cerca de 700 empresas norte-americanas que utilizam de plataformas contratuais online apresentaram necessidade com projetos de curto e longo prazo, táticos e estratégicos. Além disso, constatou que 90% destes líderes acreditam que essas plataformas serão fundamentais no futuro.

Tendo em vista essa carência, como uma plataforma poderia auxiliar esses especialistas a encontrar esses serviços de forma mais rápida e segura? Com isso, encontrou-se a necessidade de reinventar os métodos de contratação, no objetivo de facilitar a filtragem na admissão.

Sendo assim, o Search Devs vem para minimizar o problema. Baseando-se em uma aplicação web amigável para todos os usuários, sendo fácil e versátil. Como forma de inovar os métodos de contratação, seu diferencial é uma *Application Programming Interface* (API), sendo uma ferramenta que encontra a compatibilidade entre o desejo do contratante e a aptidão do contratado, tendo como consequência a agilidade no período de formação de equipes para certos projetos.

A metodologia para a busca da problemática desse projeto é fundamentada por pesquisas realizadas com livros, artigos, notícias e considerações com a vivência na atualidade. Para o desenvolvimento da aplicação será utilizado *HyperText Markup Language* (HTML5), *Cascading Style Sheet* (CSS), Bootstrap, JavaScript, JQuery, *Structured Query Language* (SQL), *Hypertext Preprocessor* (PHP) e Vetorização.

2 REFERENCIAL TEÓRICO

Para o esclarecimento do leitor, este capítulo é destinado a explicar as bases teóricas das ferramentas e tecnologias que foram utilizadas no desenvolvimento deste projeto.

2.1 HTML5

Segundo Silva (2015) o *HyperText Markup Language5* (HTML5), que em português significa Linguagem de Marcação de Hipertexto, é a versão atualizada de uma linguagem utilizada para marcação de hipertexto na criação de sites.

Hipertexto é qualquer tipo de item inserido em uma página web e tem como importante característica a função de se interligar a outros sites através dos links. Uma página web pode conter diversos itens como texto, áudio, vídeo, tabelas, imagens etc. E para criá-los o HTML marca e define a informação atribuindo-lhe significado por meio de elementos que se chamam tags. Esse significado é reconhecido por sistemas ou outros meios que possam acessar essa informação e utilizá-la (FERREIRA, 2013).

Para utilizar o HTML5 é necessário saber a estrutura básica para iniciar um projeto, que pode ser visualizada na Figura 1:

Figura 1 - Estrutura Básica HTML5

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta viewport" content="width=device-width, initial-scale=1.0">
7   <title>Index</title>
8 </head>
9 <body>
10
11 </body>
12 </html>
```

Fonte: Autoria Própria, 2022.

Segundo Eis e Ferreira (2012) as tags utilizadas na Figura 1 são definidas como a seguir:

- <!DOCTYPE html>: não é uma tag, mas uma declaração que aparece na primeira linha, ela indica os critérios que deverão ser utilizados para processar o site.

- `<html lang="en">`: a tag html é a raiz do código, ele define a estrutura e o modo que se deve seguir e está acompanhada do atributo lang que estabelece a linguagem principal do documento web, no caso da Figura 1, foi definida a sigla “en” que refere-se a *english*, o que significa que a linguagem principal é inglês.
- `<head>`: a parte inteligente do código. Aqui contém informações e metadados para utilização e controle da página web, por exemplo vinculação com outros documentos como folhas de estilo etc.
- `<meta charset="utf-8">`: essa metatag é responsável por chavear qual tabela de caracteres o site vai usar e o parâmetro utf-8 refere-se ao Unicode para utilização universal.
- `<meta http-equiv="X-UA-Compatible" content="IE=edge">`: é uma metatag para que o site consiga interpretar versões antigas do Internet Explorer (IE).
- `<meta name="viewport" content="width=device-width, initial-scale=1.0">`: essa metatag controla como o site será exibido no dispositivo.
- `<title></title>`: define o título que será apresentado na aba do navegador. □ `<body></body>`: é a tag onde todo o conteúdo da página web é desenvolvido.

O HTML é utilizado em quase todas as páginas Web que utilizamos em nosso dia a dia, e para exemplificar melhor essa tecnologia, foi desenvolvido um formulário de cadastro de usuário, que pode ser visualizado na Figura 2:

Figura 2 - Código da Tela de Cadastro

```
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta viewport" content="width=device-width, initial-scale=1.0">
8   <title>Index</title>
9
10 </head>
11
12 <body>
13   <div class="login">
14     <h1>Entrar</h1>
15
16     <form action="" method="post">
17       <input type="text" name="" id="" placeholder="Usuário">
18       <br>
19       <input type="password" name="" id="" placeholder="Senha">
20       <br>
21       <button type="submit">Entrar</button>
22     </form>
23   </div>
24
25   <div class="cadastro">
26     <h1>Cadastrar</h1>
27
28     <form action="" method="post">
29       <input type="text" name="" id="" placeholder="Usuário">
30       <br>
31       <input type="password" name="" id="" placeholder="Senha">
32       <br>
33       <button type="submit">Criar conta</button>
34     </form>
35   </div>
36 </div>
37 </div>
38 </body>
39
40 </html>
```

Fonte: Autoria Própria, 2022.

Segundo Mozilla (2021b) as tags utilizadas na Figura 2 são especificadas como a seguir:

- `<div></div>`: é um elemento não semântico no nível de bloco, que você deve usar apenas se não conseguir pensar em um elemento de bloco semântico melhor para usar ou se não quiser adicionar um significado específico.
- `<h1></h1>`: Adicione um título principal para o documento.
- `<form></form>`: Este elemento define um formulário.
- `<input>`: Insere a informação colocada no projeto.

Ao fim do desenvolvimento do código, teremos um formulário web básico, como demonstrado na Figura 3:

Figura 3 - Tela de Cadastro

Entrar

Usuário
Senha
Entrar

Cadastrar

Usuário
Senha
Criar conta

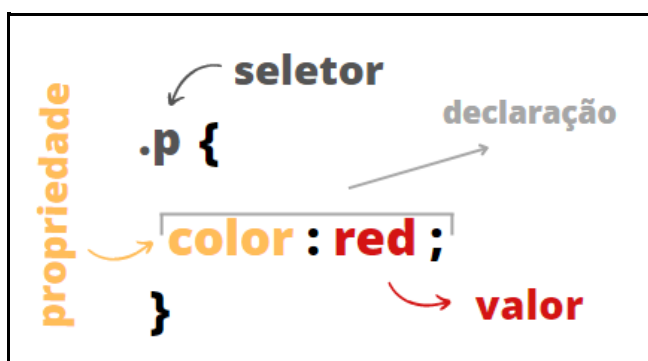
Fonte: Autoria Própria, 2022.

2.2 CSS

Cascading Style Sheet (CSS), em português, Folhas de Estilos em Cascata, segundo Silva (2008) é um instrumento simples para estilizar (por exemplo: cores, fontes, espaçamentos, tamanhos) as páginas web.

O maior objetivo do CSS é a apresentação e edição de um site. Para desenvolver a estilização é preciso usar a unidade básica da regra do CSS que é composto pelo seletor (o elemento que vai ser formatado) e declaração (determina os parâmetros manipulados na formatação). A declaração é constituída por propriedades (define a característica que será alterada do seletor) e valores (capacidade ou quantidade da propriedade) (SILVA, 2012).

Figura 4 - Estrutura Básica de CSS



Fonte: Autoria Própria, 2022.

Para que tenhamos um estilo em nossa página, é necessário aplicar o CSS ao seu documento HTML. Para isso utilizamos as tags de estilo de folha descritas na Figura 5:

Figura 5 - Tag de Estilo

```
<link rel="stylesheet" href="style.css">  
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.0-beta1/dist/css/bootstrap.min.css" rel="stylesheet">
```

Fonte: Autoria Própria, 2022.

O CSS é a tecnologia mais utilizada atualmente para desenvolver estilos em, principalmente, páginas Web. Para melhor entendimento, foi desenvolvido uma folha de estilo para o formulário exemplificado no tópico HTML5, que pode ser visualizado na Figura 6:

Figura 6 - Código de Estilo de Cadastro

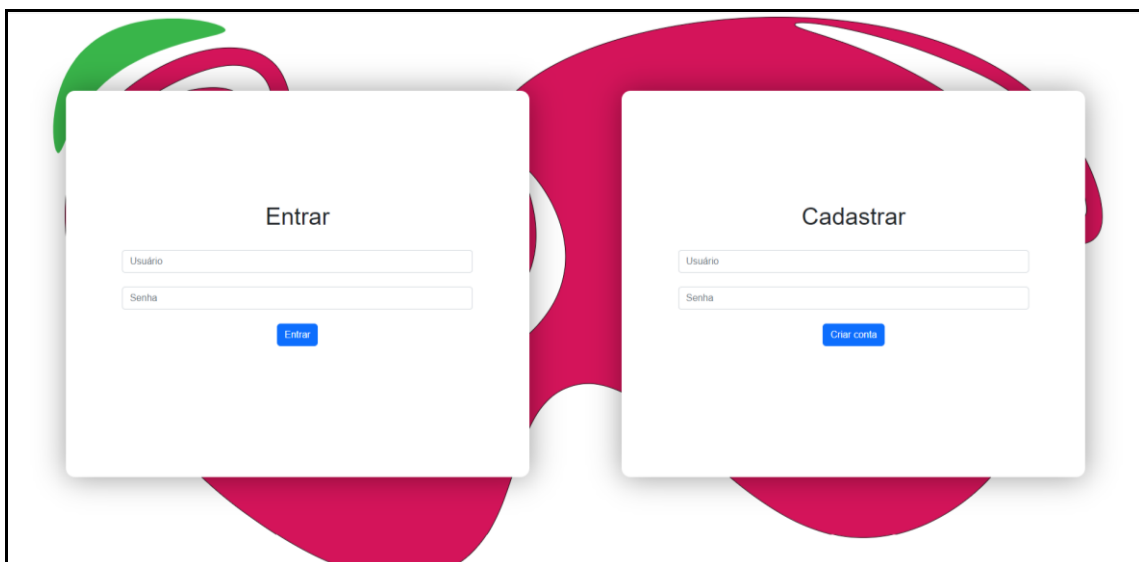
```
1  * {  
2      font-family: Arial, Helvetica, sans-serif  
3      margin: 1%;  
4  }  
5  
6  body {  
7      background-image: url('wallpaper.svg');  
8      background-position: center;  
9      background-size: 100%;  
10 }  
11  
12 .row {  
13     padding: 5% 0;  
14     align-items: center;  
15     justify-content: center;  
16     text-align: center;  
17 }  
18  
19 .login,  
20 .cadastro {  
21     background-color: #ffffff;  
22     box-shadow: 1px 1px 50px #00000067;  
23     border-radius: 15px;  
24     padding: 10%;  
25     text-align: center;  
26 }  
27  
28  
29 form {  
30     padding: 25px 10%;  
31 }  
32  
33 .login input,  
34 .cadastro input {  
35     border-radius: 5px;  
36 }
```

Fonte: Autoria Própria, 2022.

Segundo Mozilla (2021a) as tags utilizadas na Figura 6 são especificadas como a seguir:

- *Font-family*: Esta primeira regra define uma fonte base global.
- *Margin*: Define a área de margem nos quatro lados do elemento.
- *Padding*: Define uma a distância entre o conteúdo de um elemento e suas bordas.
- *Align-items*: estabelece o alinhamento de um certo item dentro do bloco que o contém.
- *Justify-content*: Define como o navegador distribui o espaço entre e ao redor dos itens de conteúdo ao longo do eixo principal de um contêiner flexível e do eixo embutido de um contêiner de grade.
- *Text-align*: Define o alinhamento horizontal do conteúdo dentro de um elemento de bloco ou caixa de célula de tabela.
- *Box-shadow*: adiciona efeitos de sombra ao redor do quadro de um elemento.
- *Border-radius*: arredonda os cantos da borda externa de um elemento.

Figura 7 - Cadastro com CSS



Fonte: Autoria Própria, 2022.

2.3 Bootstrap

O Bootstrap é um framework front-end que oferece estruturas de CSS, ou Folha de Estilo em Cascatas e Javascript para o auxílio no desenvolvimento de

aplicações web responsivas e disponibiliza uma série de componentes para criações como alertas, botões, menus, formulários etc. (LIMA, 2021).

O Bootstrap é iniciado a favor de uma tag link e tags *script* como o exemplo nas Figuras 8 e 9:

Figura 8 - tag Link Bootstrap

```
<link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/css/bootstrap.min.css" integrity="sha384-Gn5384xqQ1ao/WXA+058RXPxPg6fy4IWvTNh0E263XmFcJlSAwiGgFAW/dAiS6JXm" crossorigin="anonymous">
```

Fonte: Autoria Própria, 2022.

Figura 9 - tag Script Bootstrap

```
<script src="https://code.jquery.com/jquery-3.2.1.slim.min.js" integrity="sha384-KJ3o2DKtIkvYIK3UEENzM7KCKRr/rE9/Qpg6aAZGJwFDMVNA/GpGFF93hXpG5kkN" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.12.9/dist/umd/popper.min.js" integrity="sha384-ApNbgh9B+Y1QKtv3Rn7W3mgPxhU9K/ScQsAP7hUibx39j7fakFPskvXusvfa0b4Q" crossorigin="anonymous"></script>
<script src="https://cdn.jsdelivr.net/npm/bootstrap@4.0.0/dist/js/bootstrap.min.js" integrity="sha384-JZR6Spejh4U02d8j0t6vLEHfe/JQGiRRSQQxstfFwpi1MquVdAyjUar5+76PVCmY1" crossorigin="anonymous"></script>
```

Fonte: Autoria Própria, 2022.

O Bootstrap é um framework auxiliar ao HTML e ao CSS, ou seja, ele utiliza de elementos de ambos para seu funcionamento, e para a aplicação do mesmo, foi necessário aprimorar o formulário desenvolvido no tópico HTML5, como pode ser visualizado na Figura 10:

Figura 10 - Bootstrap na Prática

```
<body>
  <div class="container-fluid">
    <div class="row justify-content-around">
      <div class="login col-5">
        <h1>Entrar</h1>

        <form action="" method="post">
          <input type="text" name="" id="" class="form-control" placeholder="Usuário">
          <br>
          <input type="password" name="" id="" class="form-control" placeholder="Senha">
          <br>
          <button type="submit" class="btn btn-primary">Entrar</button>
        </form>
      </div>

      <div class="cadastro col-5">
        <h1>Cadastrar</h1>

        <form action="" method="post">
          <input type="text" name="" id="" class="form-control" placeholder="Usuário">
          <br>
          <input type="password" name="" id="" class="form-control" placeholder="Senha">
          <br>
          <button type="submit" class="btn btn-primary">Criar conta</button>
        </form>
      </div>
    </div>
  </div>
</body>
```

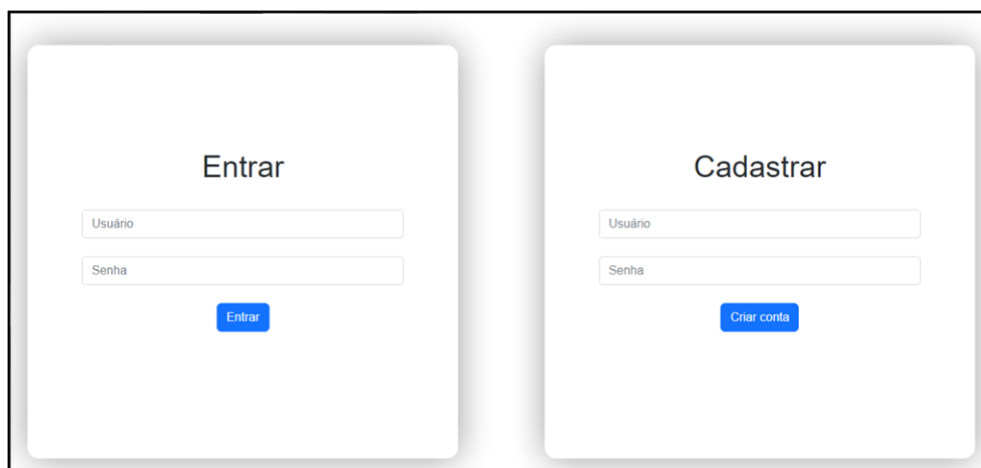
Fonte: Autoria Própria, 2022.

Segundo Bootstrap (2022) as tags utilizadas na Figura 10 são especificadas como a seguir:

- *Container-fluid*: Usado para criar um contêiner de largura total, abrangendo toda a largura da janela de visualização.
- *Row*: Utilizado para alinhar elementos em uma determinada linha. □ *Justify-content-around*: Usado para alterar o alinhamento de itens flexíveis no eixo principal.
- *Col-5*: Utilizado para delimitar a largura de uma coluna em determinada linha.
- *Form-control*: Padrão de estilo predefinido para os elementos input. □ *Btn-primary*: é a classe utilizada para atribuir a predefinição a um botão, sendo *primary* o de cor azul.

Ao fim do desenvolvimento, é possível visualizar uma página Web similar a página apresentada na Figura 11:

Figura 11 - Tela com Bootstrap



Fonte: Autoria Própria, 2022.

2.4 JavaScript

Segundo Mozilla (2021d), o JavaScript (às vezes abreviado para JS) é uma linguagem de script baseada em objetos com funções de primeira classe. Utilizada popularmente em páginas web, mas obtendo também suporte (em conjunto com pacotes de recursos externos) a plataformas Mobile e Desktop.

Foi lançado em 1995, pela Netscape em parceria com a Sun Microsystems, sendo implementada em março de 1996 no navegador Netscape *Navigator 2.0*, um dos mais utilizados navegadores do momento. De início, foi criado com o objetivo principal de rodar no lado do servidor e, com o passar do tempo, foi modificado para o modelo que temos hoje, uma linguagem *Client-Side* (SILVA, 2020, p. 23).

O Javascript é uma linguagem de programação de alto nível, ou seja, de fácil entendimento, utilizada inicialmente para adicionar interatividade ao seu site. Nos dias atuais, foram desenvolvidas novas extensões para a linguagem, a tornando muito mais poderosa e aumentando a ramificação de plataformas para aplicabilidade, como desenvolvimento *Backend*, *Desktop* e *Mobile* (MOZILLA, 2021d).

Para iniciar o desenvolvimento com Javascript, é necessário um arquivo com a extensão `.js` e um arquivo HTML para que seja exibido o seguinte resultado. Após a identificação dos arquivos, precisamos entrelaçar ambos, e para isso usamos a tag `script` (MOZILLA, 2021c), como na Figura 12:

Figura 12 - Script Javascript

```
<script src="js.js"></script>
```

Fonte: Autoria Própria, 2022.

Para exemplificar a tecnologia, foi desenvolvido um sistema de verificação de campos, tendo como base o formulário desenvolvido durante o tópico HTML5, como pode ser visualizado na Figura 13:

Figura 13 - Validação com Javascript

```
/* Javascript Vanilla */

function validarLogin() {
    var user = login.userL.value;
    var senha = login.passL.value;

    if (user == "") {
        alert('Preencha o campo com seu usuário')
        login.userL.focus;
        return false;
    } else if (senha == "") {
        alert('Preencha o campo com sua senha')
        login.passL.focus;
        return false;
    }
};

function validarCadastro() {
    var user = cadastro.userR.value;
    var senha = cadastro.passR.value;

    if (user == "") {
        alert('Preencha o campo com seu usuário')
        cadastro.userR.focus;
        return false;
    } else if (senha == "") {
        alert('Preencha o campo com sua senha')
        cadastro.passR.focus;
        return false;
    }
};
```

Fonte: Autoria Própria, 2022.

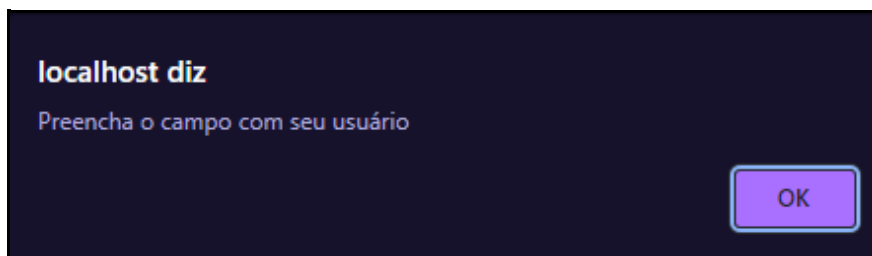
Segundo Silva (2020), as tags utilizadas na Figura 13 são especificadas como a seguir:

- `/* ... */`: Utilizado para realizar um comentário no código.
- `Function()`: Utilizado para iniciar uma função.
- `var`: Utilizado para iniciar uma variável.

- *if* (): Função de comparação, onde os parâmetros para a mesma ser verdadeira estão entre os parênteses.
- *else if*(): Função de comparação, ativo quando a primeira função retorna como “*false*”.
- *alert*(): Função utilizada para exibir um *Poup-up* com o conteúdo especificado em parênteses.
- *.value*: Um parâmetro adicional utilizado para recuperar determinado valor de um elemento.

Ao fim do desenvolvimento, o projeto com Javascript irá enviar um *alert* quando for identificado um campo sem preenchimento no formulário da Figura 14:

Figura 14 - Alerta de Javascript



Fonte: Autoria Própria, 2022.

2.5 JQuery

De acordo com Barbosa (2011), o JQuery é uma biblioteca Javascript que visa simplificar o desenvolvimento de sistemas Front-End com a linguagem, sua fama vem da possibilidade de adicionar interatividade e dinamismo com menor quantidade de linhas de códigos e a capacidade de funcionar em qualquer navegador web da atualidade.

O JQuery usa o formato $\$(selector).acao()$ para atribuir um evento a um elemento, como no exemplo na Figura 15:

Figura 15 - Exemplo de JQuery

```
/* JQuery */  
  
$(document).ready(function () {  
  //INICIAL  
  $("body").hide();  
  $("body").fadeIn(3000);  
  
  $("h1").hide();  
  $("h1").fadeIn(4000)  
  
});
```

Fonte: Autoria Própria, 2022.

As tags utilizadas na Figura 15 foram utilizadas para o desenvolvimento de animações simples dentro de uma página web, e os comandos utilizados podem ser especificados como a seguir:

- `$(document).ready(function () { ... });` : Utilizada para iniciar a leitura do código no qual é utilizado o JQuery (MOZILLA, 2021d).
- `$(“element”).hide()` : Utilizado para ocultar o elemento selecionado (SILVA, 2013, p. 306).
- `$(“element”).fadeIn()` : Utilizado para mostrar um elemento de forma suave, com um efeito *fade* (SILVA, 2013, p. 325).

2.6 Banco de Dados

O Banco de Dados pode ser definido como um conjunto de dados devidamente relacionados, basicamente uma gaveta de arquivamentos de dados computadorizados (DATE, 2004). Para um Banco de Dados é necessária uma fonte de informação, uma interação com o mundo real e um público interessado nos dados do banco (ALVES, 2013).

Projetado pela *Internacional Business Machines* (IBM), surgiu um pacote de software chamado Sistemas Gerenciadores de Banco de Dados (SGBD) para executar mais facilmente as tarefas de um banco de dados, utilizando de recursos para gerenciar os dados de uma forma mais robusta e eficiente (RAMAKRISHNAN; GEHRKE, 2007). Como exemplo de um SGBD está o MySQL, que foi usado no desenvolvimento desse projeto.

2.6.1 MySQL

De acordo com Milani (2007), o MySQL é um gerenciador de banco de dados relacional e de licença dupla, que utiliza do *Structured Query Language* (SQL) ou em português Linguagem para Pesquisas, que é muito veloz devido as implementações feitas pelos seus desenvolvedores. Além de armazenar os dados é possível usar a função de gerenciamento de acesso, integridade dos dados e relacional, transações etc., altamente portátil entre diferentes plataformas, sistemas e fornece sua API para outras linguagens como o Python, além de várias outras funções de segurança, entre outros.

Para exemplificar a funcionalidade do MySQL, foi desenvolvido uma estrutura básica de um banco de dados para usuários, como pode ser visualizado na Figura 16:

Figura 16 - Banco de Dados MySQL

```
1 CREATE DATABASE usuarios;
2 CREATE TABLE usuarios (
3     ID int NOT NULL AUTO_INCREMENT,
4     login varchar(30),
5     senha varchar(100),
6
7     PRIMARY KEY(ID)
8 );
```

Fonte: Autoria Própria, 2022.

Segundo Mysql AB (2010) as tags utilizadas na Figura 16 são especificadas como a seguir:

- **CREATE TABLE:** Cria uma tabela (tabela base), define colunas e restrições.
- **ID, login e senha:** Nomeação das tabelas.
- **Integer:** Variável que declara um número inteiro.
- **Varchar:** Variável que declara texto.
- **PRIMARY KEY():** Declara uma chave primaria.

Após a criação de nosso banco de dados, é possível inserir dados no mesmo, como visualizado na Figura 17:

Figura 17 - Cadastro no Banco de Dados

```
1 INSERT INTO usuario (login,senha) VALUES ("search-devs", "123")|
```

Fonte: Autoria Própria, 2022.

E de acordo com Mysql AB (2010), ao executá-lo teremos o resultado visualizado na Figura 18:

Figura 18 - Resultado no Banco de Dados

ID	login	senha
1	search-devs	123

Fonte: Autoria Própria, 2022.

2.7 PHP

Segundo Moraz (2005) o *Hypertext Preprocessor* (PHP) é uma linguagem voltada a criação de páginas dinâmicas que são páginas que exibem diferentes conteúdos e proporcionam interação por parte do usuário por meio de formulários, links e etc.

O PHP também serve como uma linguagem de cola (linguagens que permitem a interconexão de programas e componentes de software criados utilizando linguagens distintas), facilitando a conexão de suas páginas Web com o banco de dados do lado servidor (Converse, 2002, p.3).

Para demonstrar sua usabilidade, utilizamos o formulário utilizado no tópico HTML5 para implementação de classes de controle PHP, que podem ser demonstrados nas Figuras 19 e 20:

Figura 19 - Formulário Login PHP

```
<form action="login.php" method="post">
```

Fonte: Autoria Própria, 2022.

Figura 20 - Formulário Cadastro PHP

```
<form action="cadastro.php" method="post">
```

Fonte: Autoria Própria, 2022.

Nas classes importadas acima, serão implementadas as funções *server-side* de determinado formulário, ou seja, como iremos tratar as informações inseridas.

Esse exemplo pode ser visualizado nas Figuras 21 e 22:

Figura 21 - Cadastro PHP

```
<?php
require_once 'classe.php';
$u = new usuario();

if(isset($_POST['userR'])){
    $login = addslashes($_POST['userR']);
    $senha = addslashes($_POST['passR']);

    if(!empty($login) && !empty($senha)){
        $u->conectar('logindatabase', 'localhost', 'root', '');
        // echo "$msg";
        if ($u->msg == ""){
            if ($u->cadastar($login, $senha)){
                echo "<script language='javascript' type='text/javascript'>alert('0 usuario foi cadastrado com sucesso!')</script>";
                //echo "<script language='javascript' type='text/javascript'>>window.location.href='index.php'</script>";
            }else{
                echo "<script language='javascript' type='text/javascript'>alert('0 usuario já está cadastrado no sistema. Tente out
                echo "<script language='javascript' type='text/javascript'>>window.location.href='index.php';</script>";
            }
        }else{
            echo "Erro: ".$u->msg;
        }
    }else {
        echo "<script language='javascript' type='text/javascript'>alert('Preenha todos os campos!')</script>";
        echo "<script language='javascript' type='text/javascript'>>window.location.href='index.php'; </script>";
    }
}
```

Fonte: Autoria Própria, 2022.

Figura 22 - Login PHP

```
<?php
require_once 'classe.php';
$u = new usuario();

if(isset($_POST['userR'])){
    $login = addslashes($_POST['userR']);
    $senha = addslashes($_POST['passR']);
    /*echo "login: ".$login.", senha: ".$senha;
    echo "*/

    if(!empty($login) && !empty($senha)){
        $u->conectar('logindatabase', 'localhost', 'root', '');
        // echo "$msg: ".$msg;
        if ($u->msg == ""){
            if ($u->login($login, $senha){
                $_COOKIE['login']=$_POST['login'];
                echo "variavel global: ".$_COOKIE['login'];
                header("location: index.php");
            }else{
                echo "<script language='javascript' type='text/javascript'>alert('Não foi possível logar no sistema!')</script>";
                //echo "<script language='javascript' type='text/javascript'>>window.location.href='index.php'</script>";
            }
        }else{
            echo "Erro: ".$u->msg;
        }
    }else {
        echo "<script language='javascript' type='text/javascript'>alert('Preenha todos os campos!')</script>";
        echo "<script language='javascript' type='text/javascript'>>window.location.href='index.php'; </script>";
    }
}
```

Fonte: Autoria Própria, 2022.

De acordo com Moraz (2005), para que nossas classes de controle sejam funcionais, é necessária uma classe de modelagem. E as classes de modelagem descrevem um conjunto de dados estruturados e caracterizados por propriedades comuns.

Na Figura 23 vemos o exemplo de uma classe de modelagem:

Figura 23 - Classe de Modelagem

```
<?php
class Usuario{
    private $pdo;
    public $msg = "";

    public function conectar($dbname,$host,$usuario,$senha){
        global $pdo;
        try {
            $pdo = new PDO('mysql:host='.$host.'.dbname='.$dbname, $usuario, $senha);
        } catch (PDOException $erro) {
            global $msg;
            $msg = $erro->getMessage();
        }
    }

    public function cadastrar($login,$senha){
        global $pdo;
        $senhaMDS=MDS($senha);
        // Verifica se já existe
        $sql = $pdo -> prepare("SELECT ID FROM usuarios WHERE login = $login");
        $sql->execute();
        if($sql->rowCount() > 0){
            // já existe, cadastrar
            $sql = $pdo->prepare("INSERT INTO usuarios (login,senha) VALUES ('$login','$senhaMDS')");
            $sql->execute();
            return true;
        }
    }

    public function logar($login,$senha){
        global $pdo;
        $senhaMDS=MDS($senha);
        // // Verifica se está cadastrado
        $sql = $pdo->prepare("SELECT login FROM usuarios WHERE login = '$login' AND senha = '$senhaMDS'");
        $sql->execute();
        if($sql->rowCount() > 0){
            // está cadastrado
            $dado = $sql -> fetch();
            session_start();
            $_SESSION['ID'] = $dado['login'];
            return true;
        }else{
            // não está cadastrado
            return false;
        }
    }
}
```

Fonte: Autoria Própria, 2022.

Segundo PHP (2022) as tags utilizadas nas Figuras 21, 22 e 23 são especificadas como a seguir:

- *Class*: A definição de uma classe começa com a palavra-chave *class*, seguida do nome da classe, seguido de um par de colchetes que englobam as definições de propriedades e métodos pertencentes à classe.
- *Pdo*: função do PHP para que seja possível a conexão do banco de dados.
- *Try/Catch*: Função condicional utilizada para evitar/diagnosticar erros. □
- Global: é usada para manipular o escopo da variável, e existe o conceito de super globais em PHP, que são variáveis especiais com escopo global. □
- Return*: retorna o controle do programa para o módulo que o chamou. A execução continuará na expressão seguinte à invocação do módulo.
- *Require_Once*: é idêntica a *require* exceto que o PHP verificará se o arquivo já foi incluído, e em caso afirmativo, não o incluirá (exigirá) novamente.
- *Echo*: Exibe uma ou mais strings.
- *Empty*: Determina se a variável é vazia.
- *\$_POST*: Um array associativo de todos os campos do formulário HTML.
- *Addslashes*: Adiciona barras a uma string.

2.8 UML

O *Unified Modeling Language* (UML) é uma linguagem-padrão para criação da estrutura de projetos de software. Usada para visualização, especificação, construção e documentação de projetos com sistemas de software (FOWLER, 2014). A principal função da UML é organizar um projeto antes mesmo de iniciar o desenvolvimento e ir aprimorando com o andamento do projeto. É importante manter a documentação de software muito atualizada, pois pelo fato de ser um produto dinâmico, pode haver a necessidade de alterações por tempo indeterminado, e com uma documentação bem desenvolvida, o processo de atualização e manutenção se torna mais ágil e intuitiva (GUEDES, 2018). No UML existem vários tipos de relações entre os elementos dentro de um diagrama, mas de acordo com Stadzisz (2002) existem quatro que devem ser citadas:

- Associação: Representa que duas classes detêm uma relação entre elas (Representado por uma seta aberta).

- Inclusão: é quando um caso de uso inclui a função de outro caso de uso (representado por uma seta tracejada e possui a expressão *include* entre os sinais de maior e menor).
- Extensão: é utilizado somente em situações específicas do caso de uso, ou seja, quando uma condição é satisfeita (representado por uma seta tracejada e possui a expressão *extend* entre os sinais de maior e menor).
- Generalização: é utilizado para demonstrar a relação de um caso de uso geral e um caso mais específico (representado por uma seta fechada).

2.8.1 Requisitos de Software

De acordo com Sommerville (2013) requisitos de software são geralmente categorizados como requisitos funcionais e requisitos não funcionais:

- Requisitos funcionais: Os requisitos funcionais apontam o que ele deve fazer, dependendo do tipo de software a ser desenvolvido e dos possíveis usuários. Quando descritos como requisitos de usuário, são retratados de forma abstrata para o melhor entendimento do usuário. No entanto, sistemas mais específicos mostram essas funções em detalhes. A falta de precisão nas especificações desses requisitos é a causa de muitos problemas em software. O levantamento dos requisitos funcionais de um sistema deve ser completo e consistente.
- Requisitos não funcionais: como o nome diz, são requisitos que não estão diretamente relacionados as funções específicas do software. Os requisitos não funcionais, como desempenho ou segurança, geralmente definem as características do sistema por inteiro. É importante seguir os requisitos não funcionais, pois deixar de atender a um requisito não funcional pode inutilizar o sistema. Como exemplo, se um sistema de aeronaves não corresponder com os requisitos não funcionais como segurança e confiabilidade, não será possível operá-lo.

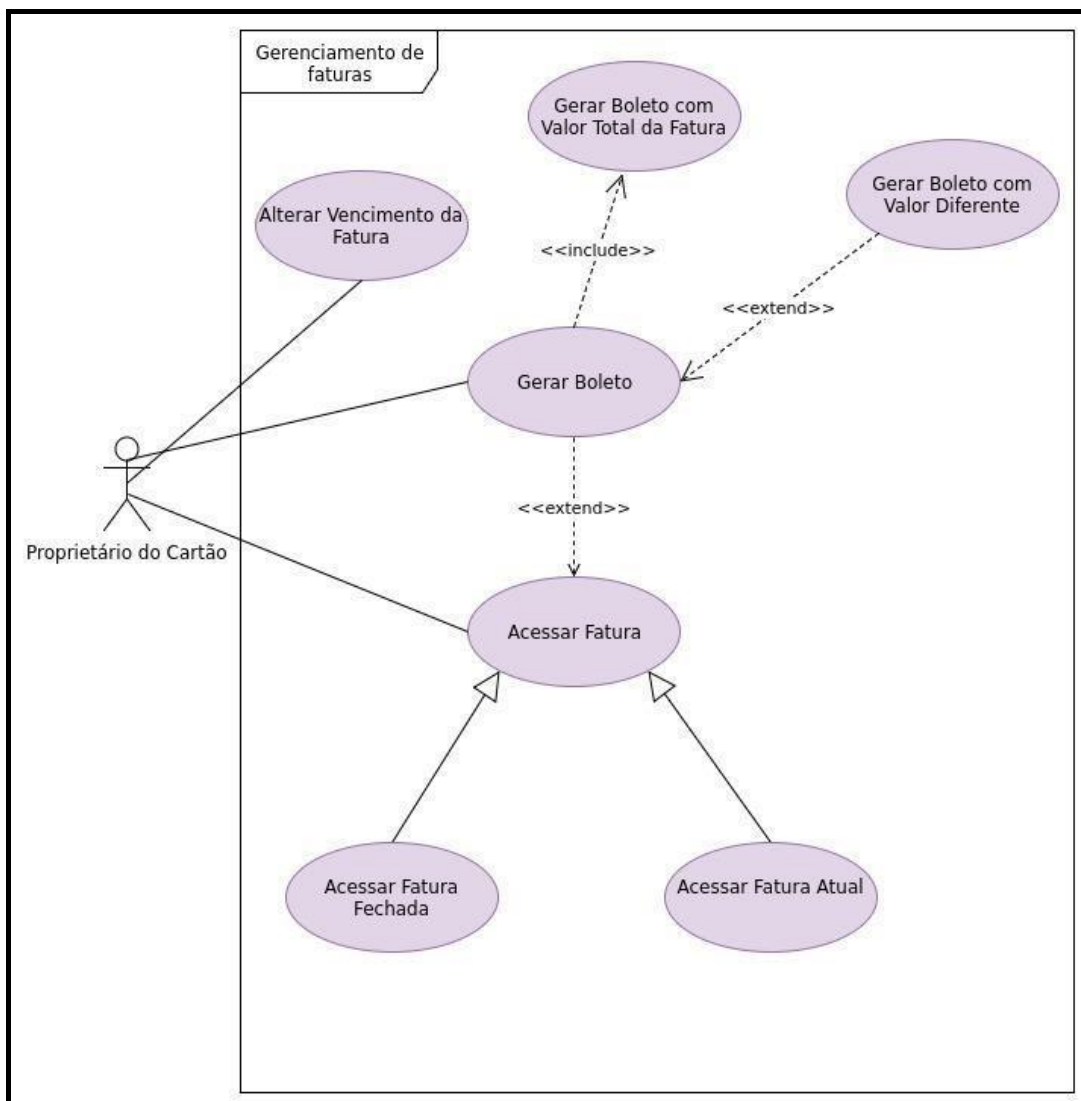
Segundo Guedes (2011) regras de negócio são normas e condições definidas pela empresa que têm de ser seguidas no processo de uma funcionalidade. Por exemplo, estabelecer que após a abertura da conta é necessário depositar um valor mínimo de início é uma regra de negócio aderida por um banco específico e que não obrigatoriamente é seguida por outras instituições bancárias.

2.8.2 Diagrama de Caso de Uso

Na Figura 24, é possível observar um diagrama de caso de uso que demonstra as possíveis interações entre um usuário do sistema e suas funcionalidades, que apresenta um cenário de gerenciamento de faturas. De acordo com Stadzisz (2002) esse diagrama é composto por relações e:

- Atores: Representam entidades externas (pessoas, dispositivos, hardwares, softwares) que interagem com o sistema. Essa interação se dá através de comunicações.
- Casos de Uso: Descreve uma aplicação ou uso completo do software

Figura 24 - Exemplo de Caso de Uso

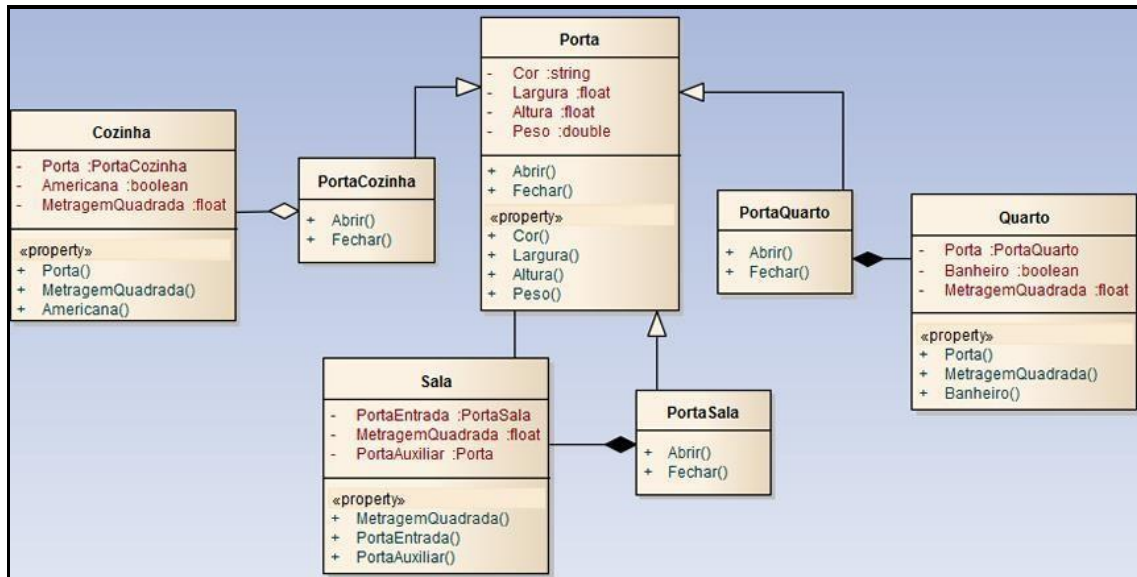


Fonte: Autoria Própria, 2022.

2.8.3 Diagrama de Classes

O Diagrama de Classes é constituído por classes, atributos e operações. A classe determina um conjunto de objetos com a mesma estrutura e comportamento, os atributos simbolizam as suas características, podendo ter visibilidade pública, privada, protegida e de pacote. As operações representam as ações e comportamento dos dados dentro do diagrama (LIMA, 2010).

Figura 25 - Exemplo de Diagrama de Classe

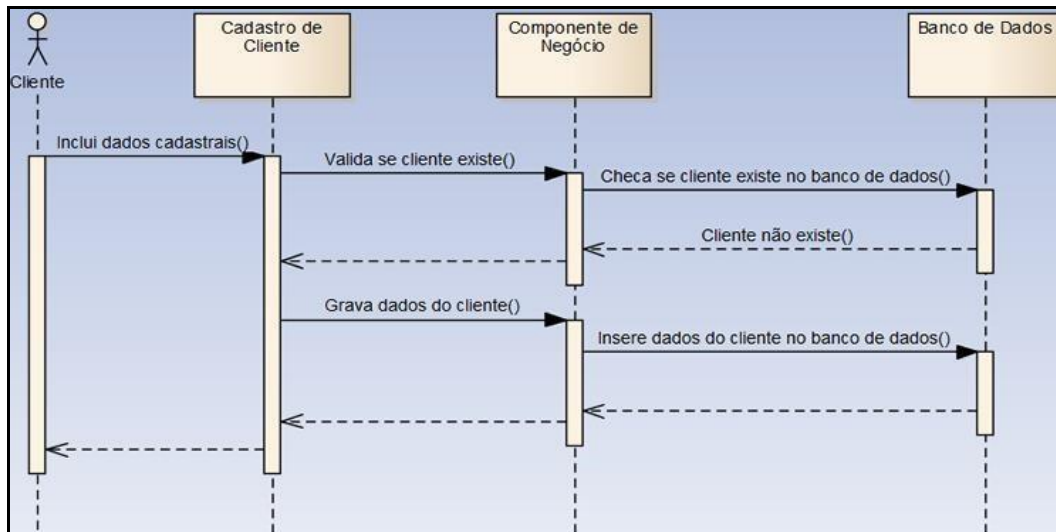


Fonte: (Ventura, 2018)

2.8.4 Diagrama de Sequência

O diagrama de sequência apresenta o caminho que as mensagens executam entre os objetos de software. Esse tipo de diagrama de interação do destaque na ordem temporal das mensagens e mostra a interação dos objetos, algo que acontece em um lugar exato do sistema (BOOCH; RUMBAUGH; JACOBSON, 2006).

Figura 26 - Exemplo de Diagrama de Sequência

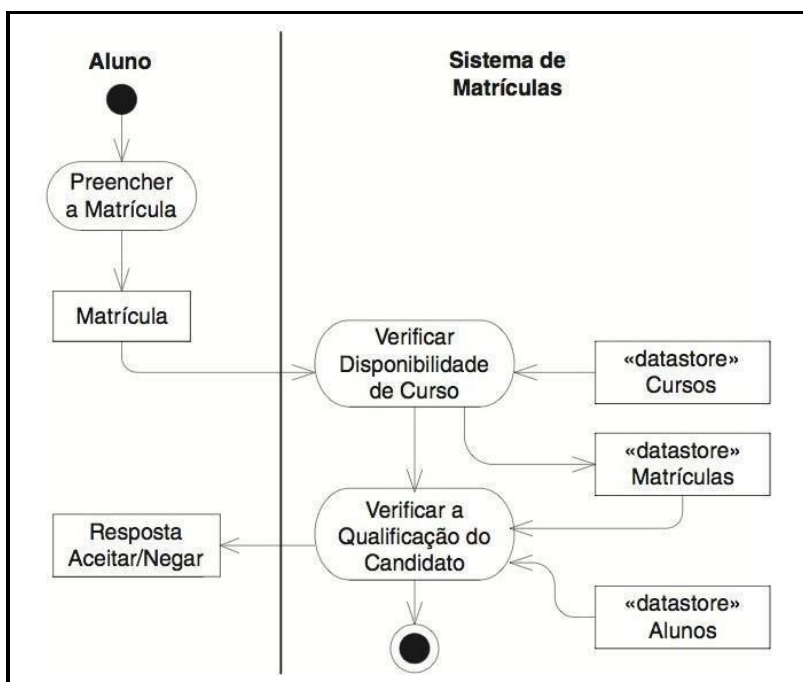


Fonte: (VENTURA, 2018)

2.8.5 Diagrama de Atividade

O diagrama de atividade apresenta uma grande quantidade de informação para mostrar uma sequência de atividades, onde existem também as atividades paralelas. O objetivo desse diagrama é pegar as ações (atividades que serão realizadas) e seus resultados em relação as variações de estados de um objeto (LARMAN, 2007).

Figura 27 - Exemplo de Diagrama de Atividade



Fonte: (RONDINA, 2015)

3 DESENVOLVIMENTO

Nesse capítulo será abordado o desenvolvimento da aplicação web Search Devs, através dos levantamos de requisitos e diagramas desenvolvidos no estudo de UML, a formação do banco de dados e a explicação de funcionalidades das telas.

3.1 Levantamento de Requisitos do Search Devs

O projeto inicial foi baseado no levantamento de requisitos funcionais, não funcionais e nas regras de negócio a partir de metas que foram definidas no começo do projeto. Na Tabela 1 é possível analisar as necessidades que foram determinados para os requisitos funcionais.

Tabela 1 - Tabela dos Requisitos Funcionais do Search Devs

Nº Requisito	Nome	Descrição
RF01	Cadastrar-se	Funcionalidade que pode ser efetuada por todos os usuários que queiram se cadastrar
RF02	Fazer Login	Funcionalidade exercida por todos os usuários para que consigam fazer todas as funções
RF03	Verificar Login	Funcionalidade exercida pelo sistema para validar a existência da conta no banco de dados
RF04	Visualizar Perfil	Funcionalidade que pode ser efetuada por qualquer usuário para que consigam ver seu próprio perfil
RF05	Editar Perfil	Funcionalidade que pode ser efetuada por qualquer usuário para que consiga alterar as suas próprias informações
RF06	Pesquisar Perfil	Funcionalidade que pode ser efetuada por qualquer usuário para que consiga buscar por perfis de outros usuários

RF07	Visualizar Novas Propostas	Funcionalidade disponível para desenvolvedores onde podem checar se existem novas propostas de trabalho
RF08	Visualizar Projetos em Andamento	Funcionalidade que pode ser efetuada por todos os usuários para que consigam visualizar quais projetos participam/administram
RF09	Visualizar Projetos	Funcionalidade que pode ser efetuada por qualquer usuário para que consiga visualizar detalhes do projeto que participa/administra
RF10	Criar Projetos	Funcionalidade disponível para contratantes para que solicitem o desenvolvimento de um projeto
RF11	Editar Projetos	Funcionalidade executada por contratantes para alterar informações e necessidades do projeto que administram
RF12	Aceitar/Negar Proposta	Funcionalidade que pode ser efetuada por desenvolvedores para que ingressem ou não no projeto que foi encaminhado para ele
RF13	Aceitar/Negar profissional	Funcionalidade que pode ser efetuada por contratantes para que consigam definir qual desenvolvedor pode ingressar no projeto e assim iniciar o desenvolvimento
RF14	Filtragem de Perfis	Funcionalidade executada pelo sistema para que ele colete as informações, filtre e defina qual o projeto ideal para o desenvolvedor e qual o desenvolvedor ideal para o projeto

RF15	Enviar Notificações	Funcionalidade executada pelo sistema para enviar novas propostas para desenvolvedores quando projetos forem criados
------	---------------------	--

Fonte: Autoria Própria, 2022.

Na Tabela 2, é apresentado os requisitos não funcionais que foram estabelecidos antes do início do desenvolvimento do projeto.

Tabela 2 - Tabela dos Requisitos Não Funcionais do Search Devs

Nº Requisitos Não Funcionais	Categoria	Descrição
RNF01	Intuitivo(amigável)	Ter uma boa interface para melhor interação entre o usuário e o sistema
RNF02	Linguagem	Foram utilizadas as linguagens PHP, HTML5, CSS e JavaScript
RNF03	Banco de Dados	O banco de dados utilizado foi o MySQL
RNF04	Conexão	Necessária conexão com a internet para utilizar
RNF05	Plataforma	Plataforma web tanto para desktop quanto para mobile
RNF06	Agilidade	Rapidez na resposta da IA
RNF07	Hospedagem	LocaWeb
RNF08	Servidor	Linux

Fonte: Autoria Própria, 2022.

A tabela 3, demonstra as regras de negócio que foram definidas para que fosse possível começar o desenvolvimento do projeto.

Tabela 3 - Tabela das Regras de Negócio do Search Devs

Nº Regras de Negócios	Descrição

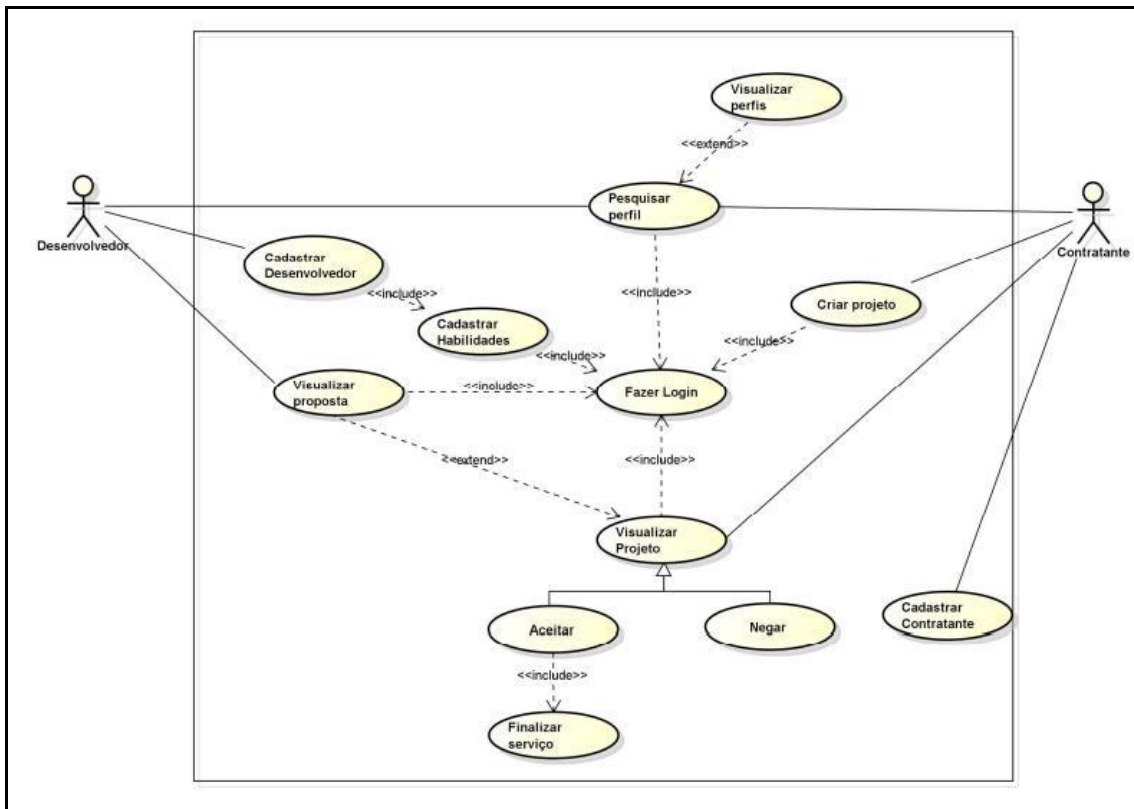
RN01	Funcionalidades específicas somente após Login
RN02	Somente empresas podem criar projetos
RN03	Somente freelancers podem ingressar em projetos
RN04	Há a necessidade de concordância e aceitação de ambas as partes envolvidas para iniciar um projeto
RN05	Usuários menores que 16 anos não têm acesso a plataforma
RN06	O CPF e o CNPJ cadastrados têm de ter a existência validada
RN07	Para terminar o cadastro, desenvolvedores devem cadastrar suas áreas de atuações e suas habilidades
RN08	O sistema deverá ranquear os desenvolvedores do que mais tem habilidades adequadas para o que tem menos
RN09	É necessário que o usuário aceite os termos de uso informados no cadastro para que utilize do site

Fonte: Autoria Própria, 2022.

3.2 Diagrama de Caso de Uso do Search Devs

O diagrama de caso de uso apresentado na Figura 28, demonstra a interação dos usuários desenvolvedor e contratante com o sistema, apresentando as ações que poderão ser feitas na aplicação por meio de casos de uso e relações. Observando-o é possível perceber que é um projeto que possui o propósito de contratação de serviços e que além do cadastro, todas as funções tem a obrigatoriedade de realizar o login antecipadamente.

Figura 28 - Diagrama de Caso de Uso do Search Devs



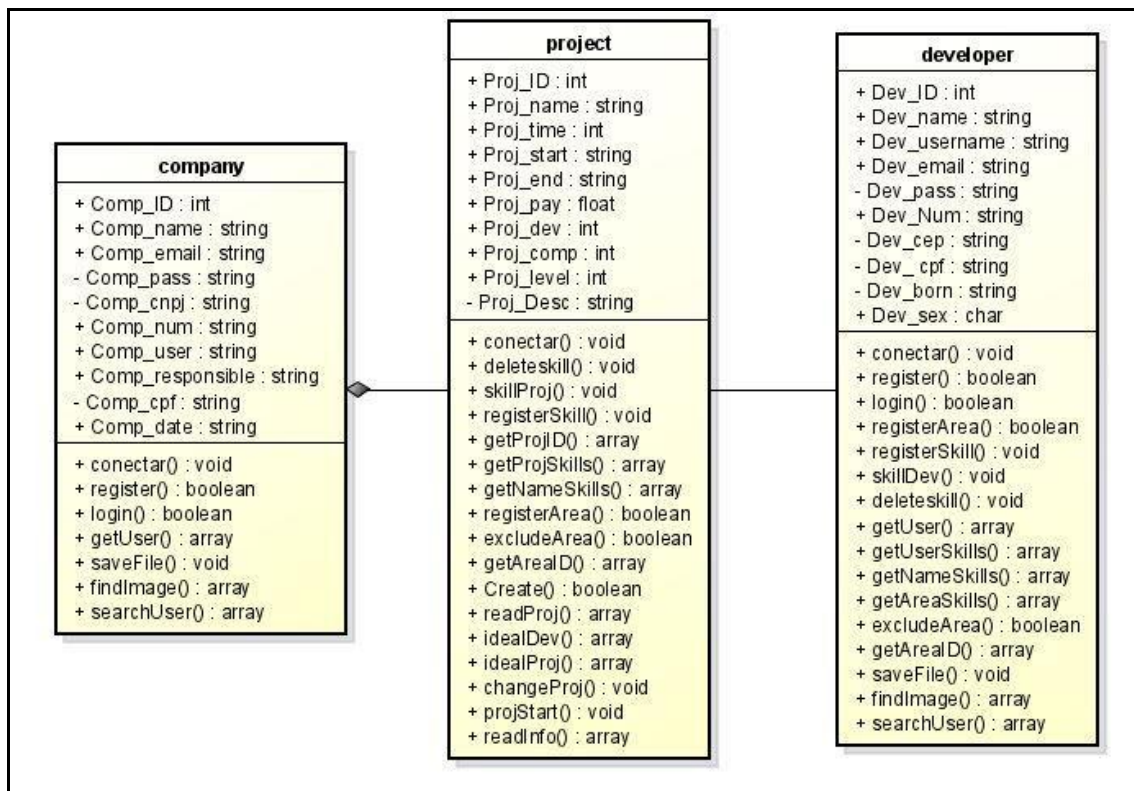
Fonte: Autoria Própria, 2022.

Baseando-se no diagrama de caso de uso, foram desenvolvidos todos os outros diagramas do projeto.

3.3 Diagrama de classes do Search Devs

O diagrama de classes apresentado na Figura 29, representa a interação entre as classes existentes no projeto.

Figura 29 - Diagrama de classes do Projeto Search Devs



Fonte: Autoria Própria, 2022.

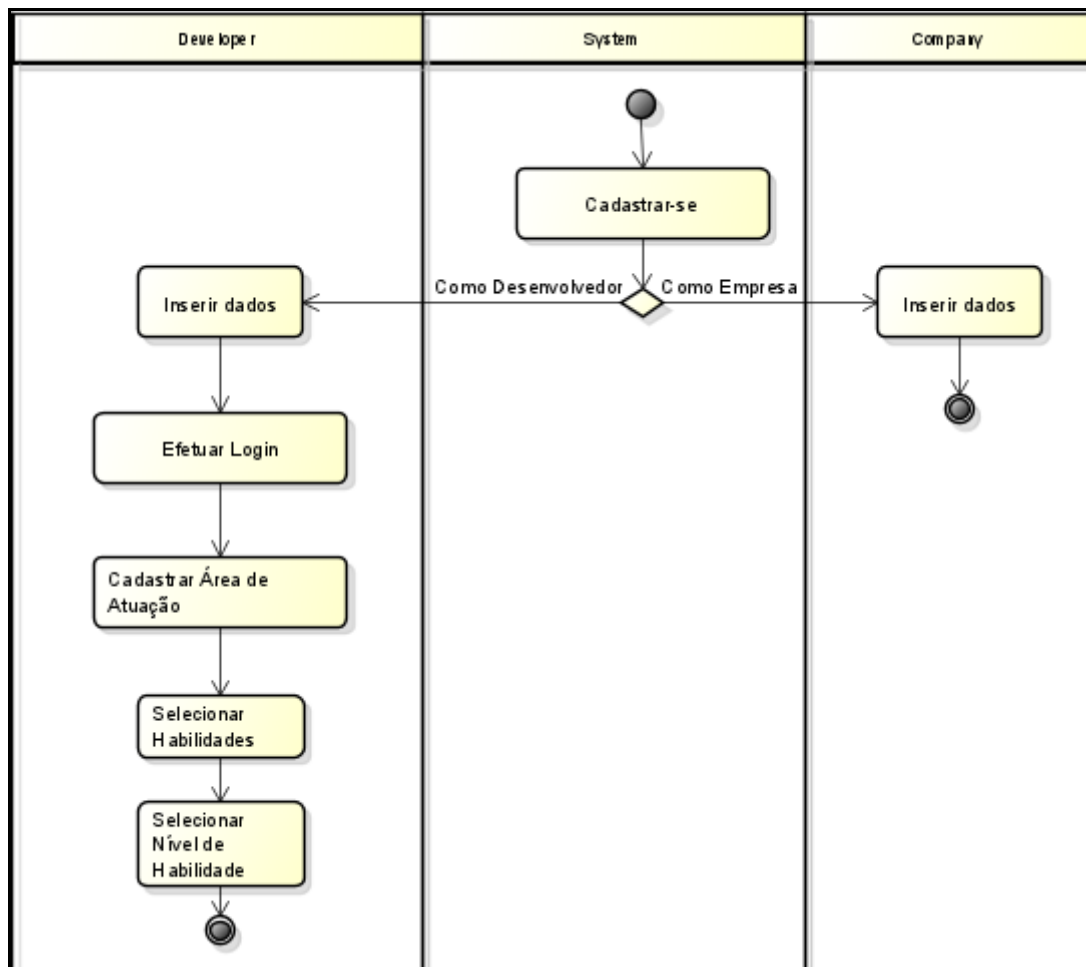
A classe *developer* tem uma relação de associação com a classe *project*, pois o objeto *Dev_ID* associa-se ao objeto *Proj_Dev*. A classe *company* tem uma relação de agregação com a classe *project* porque ela não pode existir sem a classe *company*.

3.4 Diagrama de atividades do Search Devs

Os diagramas de atividades foram divididos em três, os quais demonstram as atividades efetuar login, efetuar cadastro e as atividades gerais do site.

A Figura 30 apresenta, através das atividades, o fluxo que deve ser seguido para efetuar o cadastro.

Figura 30 - Diagrama de Atividades do Cadastro

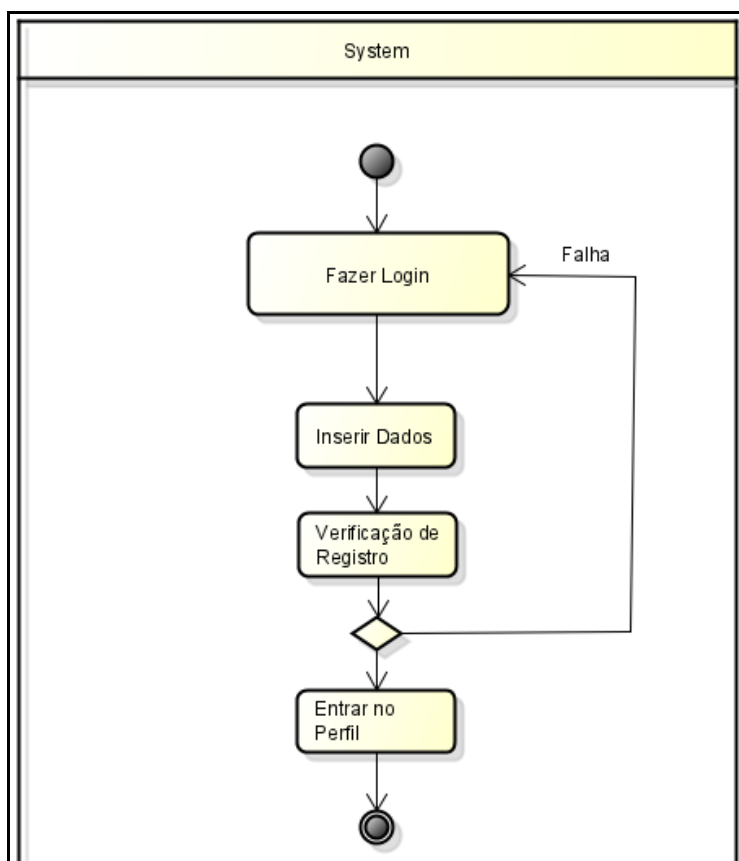


Fonte: Autoria Própria, 2022.

O fluxo é iniciado na página de cadastro onde haverá duas escolhas: se cadastrar como desenvolvedor ou como empresa. Ao se cadastrar como desenvolvedor, além de, colocar informações básicas como no cadastro de empresa, deverá também informar a área de atuação no mercado de trabalho tecnológico, informando quais habilidades possui e em que nível está.

Na Figura 31, é possível observar o fluxo que deve ser seguido para execução do login.

Figura 31 - Diagrama de Atividades Login

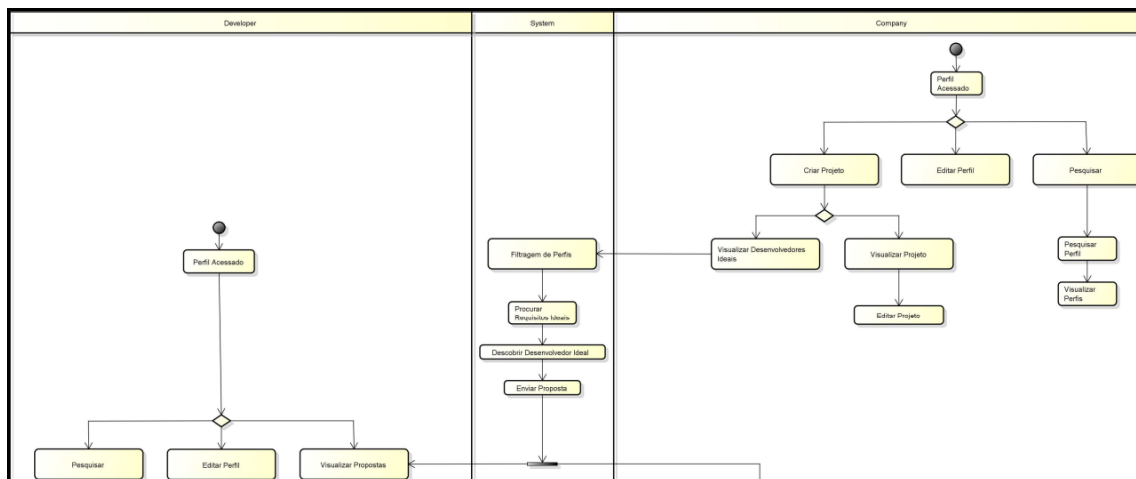


Fonte: Autoria Própria, 2022.

O fluxo é iniciado a partir da atividade fazer login, para efetuá-lo é necessário inserir os dados que já foram cadastrados anteriormente. Após inseridos, será feita a verificação de registro é feita através da busca das informações no banco de dados, se falhar, o usuário retorna a página de login, se funcionar, é direcionado ao seu próprio perfil.

Nas Figuras 32 e 33 são mostradas as atividades que podem ser feitas partir do momento em que o cadastro e o login foram executados.

Figura 32 - Diagrama de Atividades Gerais 1



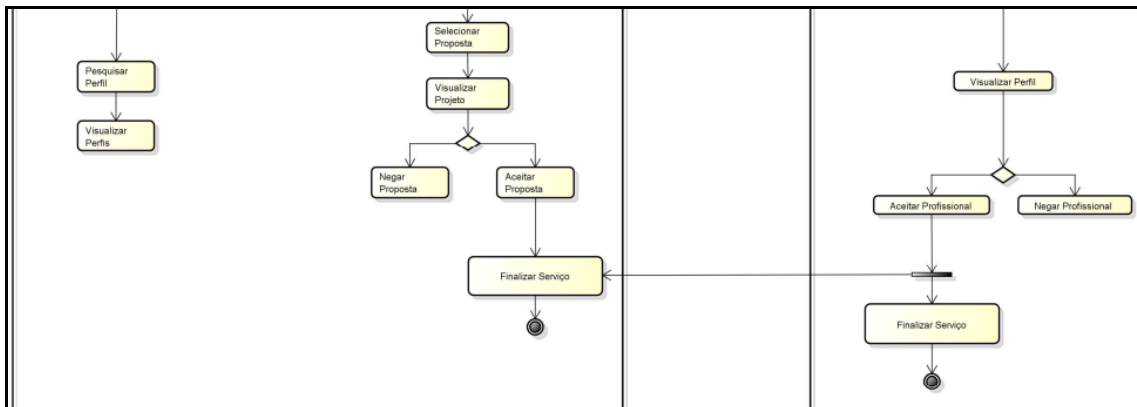
Fonte: Autoria Própria, 2022.

O fluxo da empresa inicia-se a partir do momento que está logado. O usuário empresa pode escolher entre pesquisar perfil, onde fará a busca de perfis e poderá visualizar informações de outros profissionais e contratantes, editar perfil, no qual será capaz de alterar as informações em seu próprio perfil ou criar um projeto inserindo as necessidades e informações do projeto, a partir disso a empresa conseguirá visualizar os profissionais ideais para o seu projeto ou editar as informações e requisitos do projeto.

O sistema pegará as informações do serviço e checará quais são as habilidades que o projeto necessita, assim fará uma filtragem com os desenvolvedores cadastrados no banco de dados, irá procurar por profissionais que tenham as habilidades ideais para o trabalho e encaminhará uma proposta tanto para o desenvolvedor quanto para o contratante.

O desenvolvedor inicia seu caminho quando faz login, assim como a empresa, o desenvolvedor pode pesquisar por perfis de profissionais e contratantes e editar os seus dados de perfil. A outra ação que pode ser executada por ele é a partir do encaminhamento da proposta feita pelo sistema. Ele pode checar as propostas, selecioná-las, visualizar o projeto e escolher se aceita ou nega a oferta de trabalho, se negarem podem ir escolher entre outros projetos que serão enviados se suas habilidades se adequarem, se aceitarem, então terá que aguardar a resposta da empresa, se os dois se aceitarem logo o desenvolvimento do projeto começa e ambos têm que finalizá-lo.

Figura 33 - Diagrama de Atividades Gerais 2



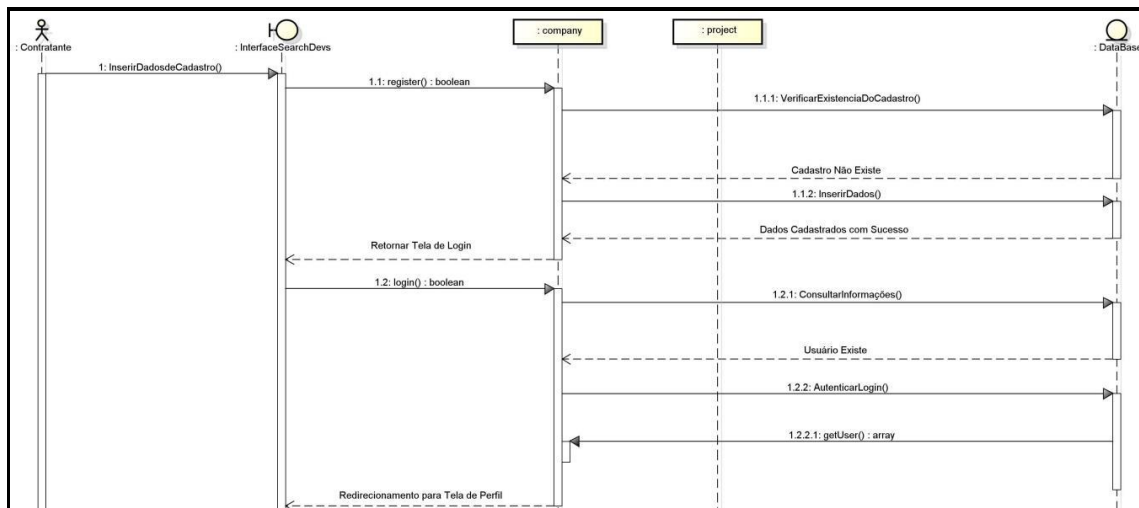
Fonte: Autoria Própria, 2022.

3.5 Diagrama de Sequência do Search Devs

O diagrama de sequência compõe no total dois diagramas, os quais são funcionalidades gerais da aplicação para o contratante e outro para as funcionalidades gerais do desenvolvedor.

Na Figura 34, o diagrama descreve o processo de realização do cadastro por parte do contratante, demonstrando o percurso que ocorre entre a aplicação e o banco.

Figura 34 - Diagrama de Sequência Cadastro e Login do Contratante



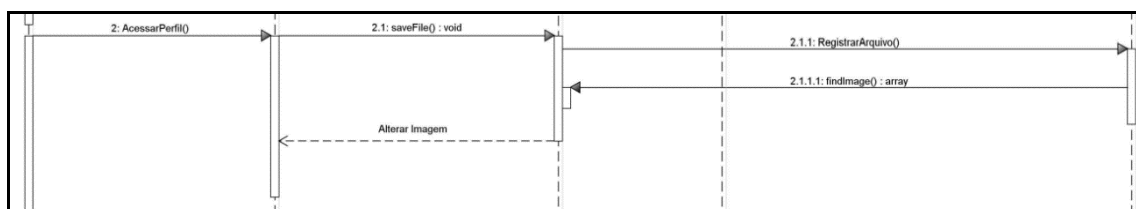
Fonte: Autoria Própria, 2022.

A sequência inicia-se quando o usuário acessa a página de criação de perfil, insere as suas informações profissionais através do método `register` chamado pela classe `company`. Após a verificação de duplicidade de cadastros, as

informações do usuário são registradas no banco de dados. Depois desse processo já é possível acessar a plataforma utilizando as informações precedentes. Após o fim do processo de cadastro o usuário é encaminhado a página de login através do método *login* para que consiga colocar as informações e entrar em sua conta no site. Para isso, há a verificação de dados para checar se houve ou não um cadastro anteriormente, e se houver, então o usuário conseguirá acessar o seu perfil.

Na Figura 35, o diagrama demonstra a alteração de imagem do perfil do usuário contratante.

Figura 35 - Diagrama de Sequência de Alteração de Imagem do Contratante

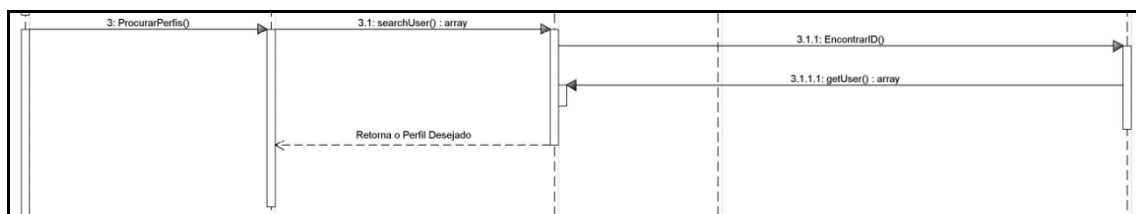


Fonte: Autoria Própria, 2022.

A Figura 35, mostra quando o contratante acessa o perfil e através do método *saveFile*, ele inicia o processo de definir e registrar no banco de dados a imagem que foi escolhida através do método *findImage*, alterando a imagem.

Continuando na Figura 36, o contratante pode pesquisar os usuários pelo método *searchUser*, então ocorrerá a busca pelo ID no banco de dados e através do método *getUser*, será encaminhado o perfil que foi buscado pelo contratante.

Figura 36 - Diagrama de Sequência de Pesquisa de Perfis do Contratante

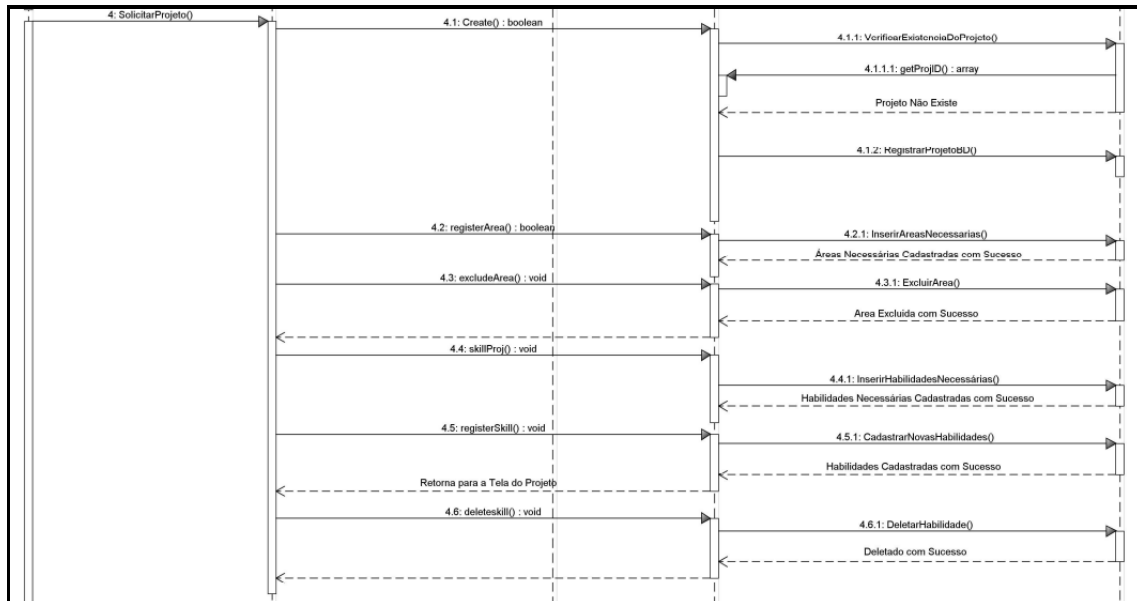


Fonte: Autoria Própria, 2022.

Na Figura 37, é apresentado o processo de criação do projeto pelo contratante, que inicia por meio do método *Create*, onde primeiro será verificado se já existe um projeto igual e se não existir poderá haver a criação do projeto. A partir disso, o contratante poderá cadastrar as informações do projeto pelos métodos

registerArea (cadastra as áreas que são necessárias para a execução do projeto), *excluiArea* (excluir as áreas que não vão ser utilizadas no projeto), *skillProj* (cadastra as habilidades necessárias que o desenvolvedor tem que ter para participar do desenvolvimento do trabalho), *registerSkill* (cadastra novas habilidades que não foram pré-definidas pelo site, mas são necessárias para o projeto) e *deleteSkill* (deleta as habilidades que o projeto não necessita mais).

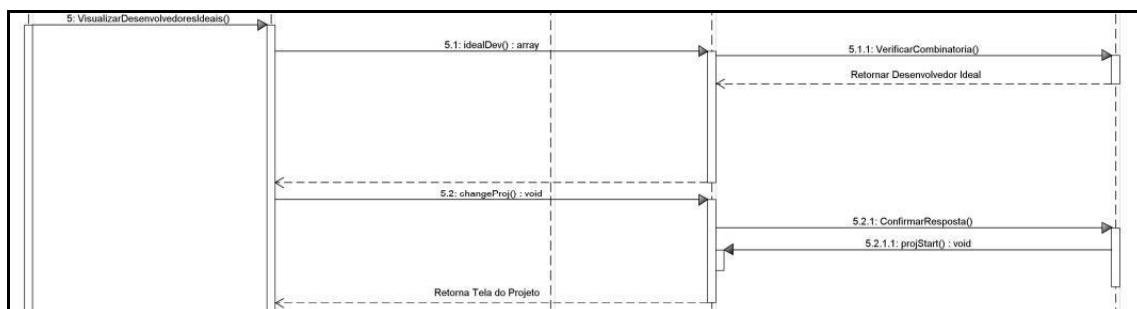
Figura 37 - Diagrama de Sequência da Criação de Projeto do Contratante



Fonte: Autoria Própria, 2022.

Na Figura 38, é demonstrado a parte de escolher o desenvolvedor ideal para o projeto do contratante.

Figura 38 - Diagrama de Sequência Visualizar Desenvolvedor Contratante

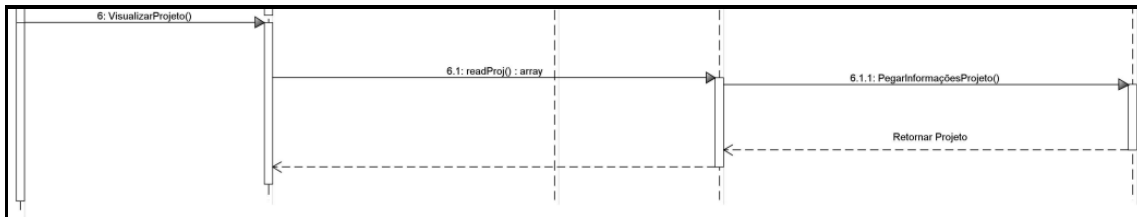


Fonte: Autoria Própria, 2022.

O contratante acessa a lista de desenvolvedores ideais, nesse momento a inteligência artificial faz a combinação de áreas e habilidades e retorna o profissional que tem o necessário para a execução do projeto, então a empresa

aceita o profissional e aguarda sua resposta para que assim, iniciem seu projeto através do método *projStart*.

Figura 39 - Diagrama de Sequência Visualizar Projeto do Contratante

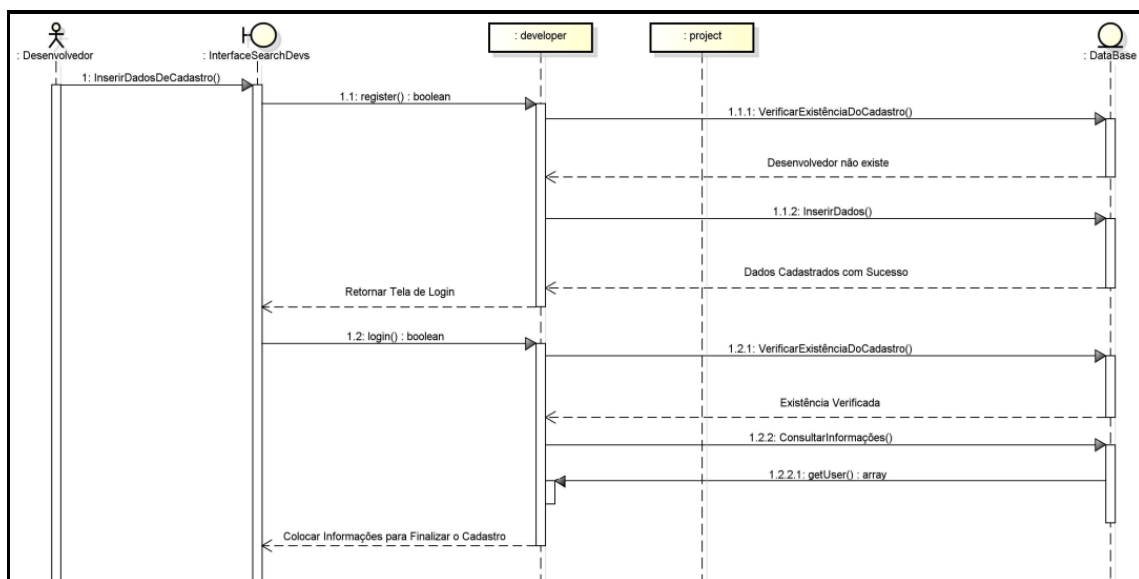


Fonte: Autoria Própria, 2022.

O diagrama na Figura 39, explica como funciona a visualização do projeto, que ocorre pelo método *readProj*, pega as informações no banco de dados e encaminha para o contratante.

A Figura 40 mostra a parte de cadastro e login do desenvolvedor no diagrama.

Figura 40 - Diagrama de Sequência do Cadastro e Login do desenvolvedor

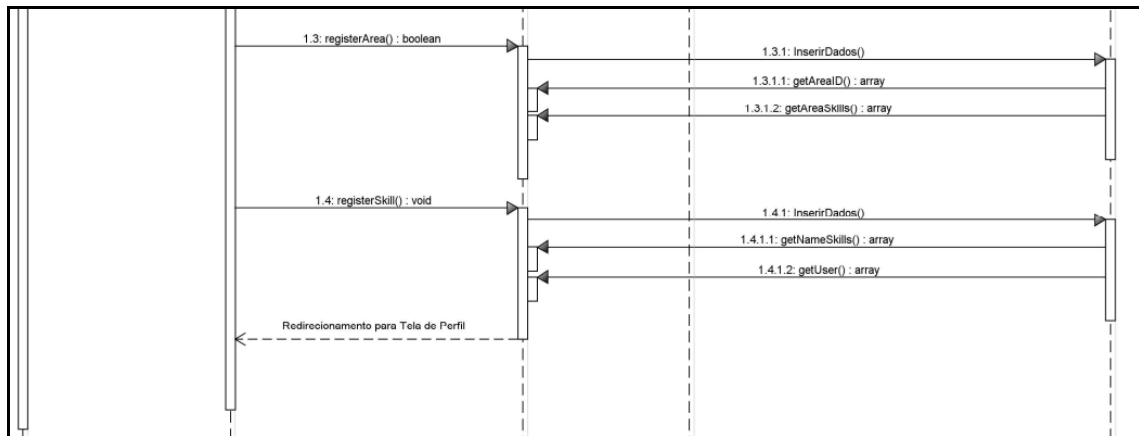


Fonte: Autoria Própria, 2022.

A sequência inicia quando o usuário acessa o site para se cadastrar, insere as suas informações pessoais através do método *register* chamado pela classe *developer*. Após a verificação de cadastros para checar se já existe uma conta igual, as informações do usuário são registradas no banco de dados. Após o fim do cadastro o usuário é encaminhado a página de login através do método *login* para que consiga colocar as informações e entrar em sua conta no site. Para isso, há a verificação de dados para checar se houve ou não um cadastro

anteriormente, e se houver, então o usuário conseguirá acessar o seu perfil. Diferente do contratante, o desenvolvedor deverá colocar sua área de atuação (através do método *registerArea*) e suas habilidades (através do método *registerSkill*) para que possa realmente finalizar o cadastro e utilizar o site em sua forma total.

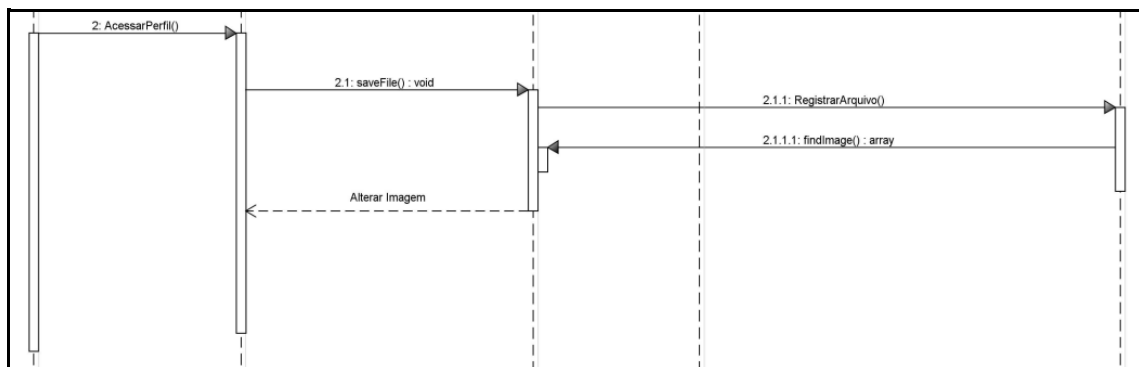
Figura 41 - Diagrama de Sequência do Cadastro e Login do Desenvolvedor 2



Fonte: Autoria Própria, 2022.

A Figura 42, demonstra a parte de editar a imagem do perfil do desenvolvedor.

Figura 42 - Diagrama de Sequência Alteração de Imagem Desenvolvedor

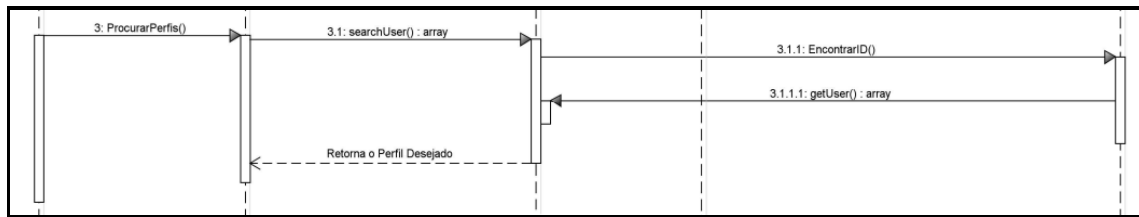


Fonte: Autoria Própria, 2022.

A Figura 42, demonstra quando o desenvolvedor entra em seu perfil e através do método *saveFile*, do mesmo modo que o contratante, ele inicia o processo de definir e registrar no banco de dados a imagem que foi escolhida através do método *findImage*, alterando a imagem.

Na Figura 43, é possível ver a sequência que ocorre para que o desenvolvedor consiga pesquisar por outros perfis.

Figura 43 - Diagrama de Sequência de Pesquisa Perfil Desenvolvedor

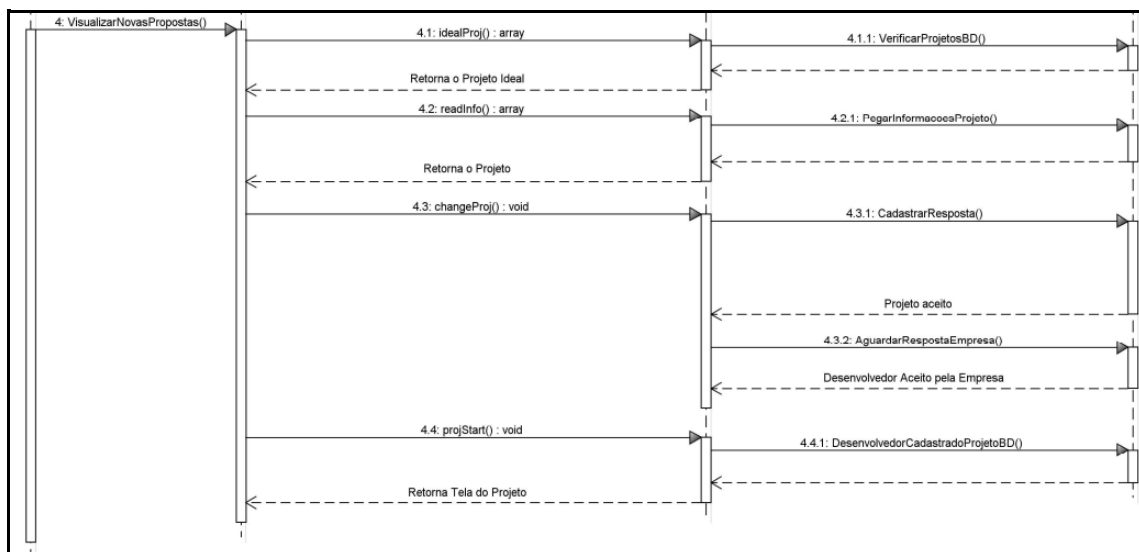


Fonte: Autoria Própria, 2022.

Na Figura 43, o contratante assim como desenvolvedor pode buscar os usuários pelo método `searchUser`, então irá ocorrer a busca pelo ID no banco de dados e através do método `getUser`, será encaminhado o perfil que foi buscado pelo contratante.

Na Figura 44, é mostrada a sequência que ocorre na visualização de propostas para o usuário desenvolvedor.

Figura 44 - Diagrama de Sequência Visualizar Propostas Desenvolvedor



Fonte: Autoria Própria, 2022.

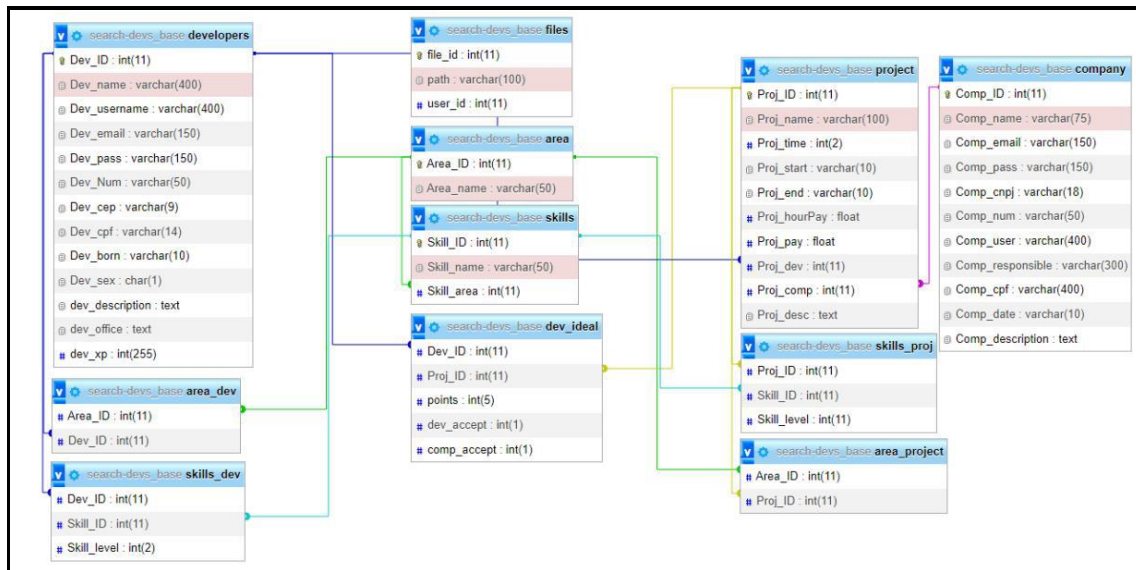
O desenvolvedor acessa a área de novas propostas, então começa o processo de verificação para achar projetos que se encaixe em suas habilidades cadastradas anteriormente. Pelo método `readInfo`, se torna capaz ler as informações do projeto quando encaminhado para o desenvolvedor. Depois de ler os dados do projeto, o desenvolvedor vai decidir se aceita ou não participar do projeto através do método `changeProj`, que vai cadastrar a resposta no banco de dados e aguardar a resposta da empresa. Quando a empresa aceitar o

desenvolvedor o projeto começa através do método *projStart* retorna a tela do projeto para o desenvolvedor.

3.6 DER

O diagrama entidade relacionamento da aplicação Search Devs é formada por onze entidades, as quais estão diretamente ou indiretamente relacionadas.

Figura 45 - DER do Search Devs

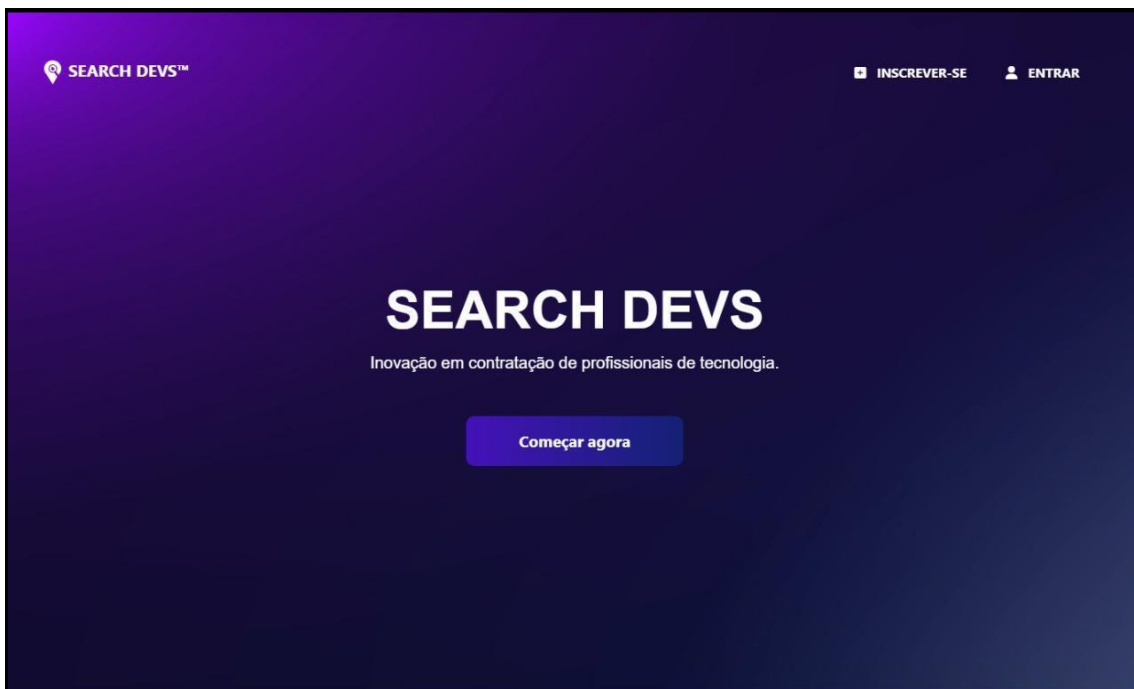


Fonte: Autoria Própria, 2022.

3.7 Telas Search Devs

Ao acessar o site do Search Devs, será possível visualizar a tela ilustrada na Figura 46.

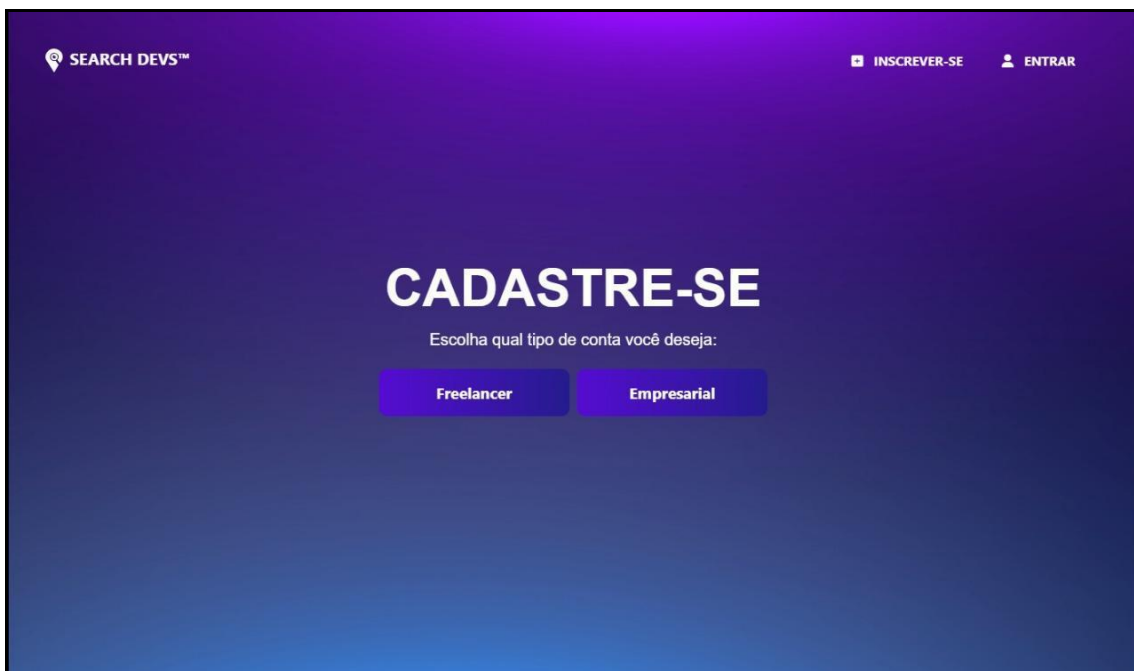
Figura 46 - Home



Fonte: Autoria Própria, 2022.

Quando clicar no botão começar agora, o usuário será direcionado a página de cadastro demonstrada na Figura 47.

Figura 47 - Tela de escolha de Cadastro

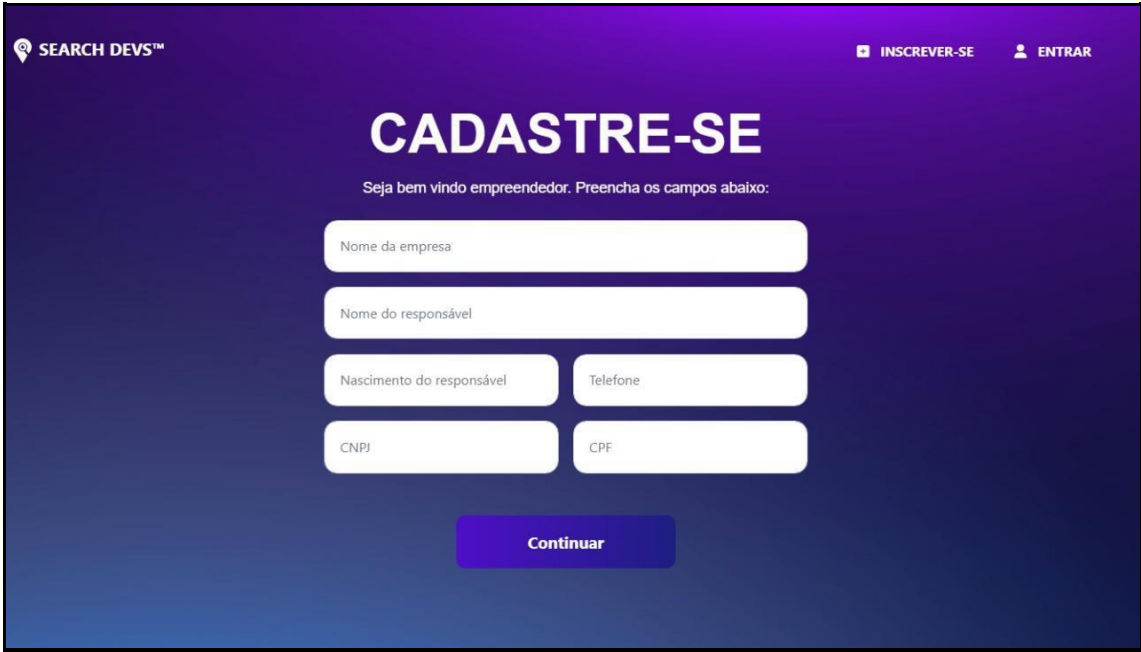


Fonte: Autoria Própria, 2022.

A primeira etapa do cadastro é escolher entre ser desenvolvedor ou empresa. Iniciado o processo, o usuário será direcionado para continuar o cadastro de acordo com a opção decidida.

A tela que pode ser visualizada na Figura 48, é a continuação do cadastro de um usuário que escolheu se inscrever como um empreendedor. É necessário que preencha o formulário de acordo com os dados de sua empresa.

Figura 48 - Tela de Cadastro Empreendedor



A imagem mostra a interface de usuário para o cadastro de um empreendedor. No topo, há o logotipo "SEARCH DEVS™" e dois links: "INSCREVER-SE" e "ENTRAR". O título principal é "CADASTRE-SE" em letras brancas grandes. Abaixo dele, uma mensagem diz: "Seja bem vindo empreendedor. Preencha os campos abaixo:". O formulário contém seis campos de entrada: "Nome da empresa", "Nome do responsável", "Nascimento do responsável", "Telefone", "CNPJ" e "CPF". Um botão azul com o texto "Continuar" está centralizado na base do formulário.

Fonte: Autoria Própria, 2022.

A tela apresentada na Figura 49, é a continuação do cadastro de um usuário que escolheu se inscrever como um desenvolvedor. É necessário que preencha o formulário de acordo com os seus dados pessoais.

Figura 49 - Tela de Cadastro Desenvolvedor

The screenshot shows a registration form titled 'CADASTRE-SE' on a dark purple background. At the top left is the 'SEARCH DEVS™' logo, and at the top right are links for 'INSCREVER-SE' and 'ENTRAR'. The main heading is 'CADASTRE-SE' in large white letters, followed by the instruction 'Seja bem vindo desenvolvedor. Preencha os campos abaixo:'. The form contains several input fields: 'Nome Completo', 'Data de Nascimento', 'Telefone', 'CEP', and 'CPF'. Below these fields are radio buttons for 'Sexo' with options 'Feminino', 'Masculino' (selected), and 'Outro'. A blue 'Continuar' button is centered at the bottom of the form.

Fonte: Autoria Própria, 2022.

Na figura 50 temos a continuação do cadastro do desenvolvedor. Nesta tela serão registradas as áreas de atuação do desenvolvedor para que os projetos correspondam.

Figura 50 - Tela de Cadastro Área

The screenshot shows a page titled 'FINALIZAR CADASTRO' on a dark purple background. At the top left is the 'SEARCH DEVS™' logo, and at the top right is a 'SAIR' link. The main heading is 'FINALIZAR CADASTRO' in large white letters, followed by 'Bem vindo, desenvolvedor.' and the instruction 'Por favor, selecione sua(s) área(s) de atuação:'. Below this is a grid of checkboxes for various development areas. The selected areas are 'Blockchain', 'Web - Back End', and 'Web - Front End'. Other areas include 'Artificial Intelligence', 'Database Analytics', 'Desktop Development', 'Data Science', 'DevOps', 'Cybersecurity', 'Design', 'Mobile Development', and 'Outros'. A blue 'Continuar' button is centered at the bottom of the page.

Fonte: Autoria Própria, 2022.

A finalização de cadastro do desenvolvedor está retratada na figura 51. Nesta etapa o desenvolvedor irá adicionar habilidades ao seu perfil e o nível delas.

Figura 51 - Tela de Cadastro Desenvolvedor Habilidades

SEARCH DEVS™ SAIR

FINALIZAÇÃO DE CADASTRO

Bem vindo, desenvolvedor.

Por favor, selecione sua(s) habilidades:

Ex: HTML5 Área de Atuação Nível: 1 2 3 4 5 ✓

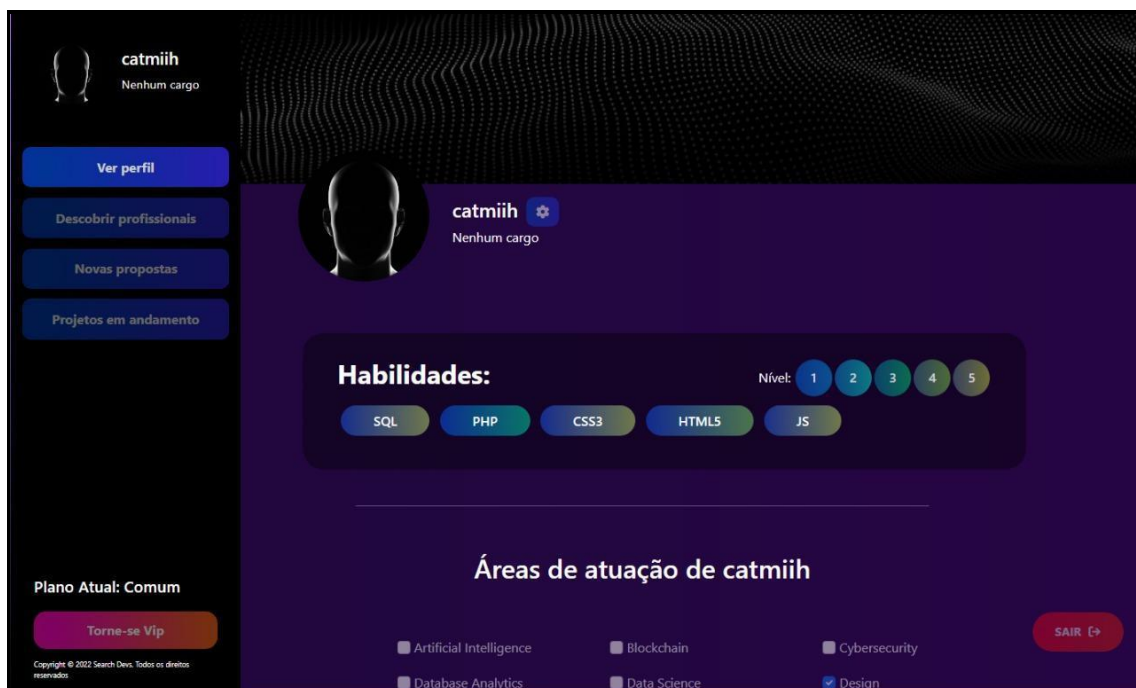
× HTML5 × CSS3 × PHP × SQL × JavaScript

Começar agora

Fonte: Autoria Própria, 2022.

A Figura 52 exibe o perfil de um usuário cadastrado como desenvolvedor. Nele é possível visualizar as habilidades e os níveis do desenvolvedor, suas áreas de atuação, informações profissionais como nome e telefone. Também é capaz de pesquisar outros perfis pressionando o botão descobrir profissionais que fica no canto esquerdo da tela com outros botões, como as novas propostas, que serve para visualizar novos projetos que foram encaminhados ao usuário e por fim, visualizar os projetos que já está participando.

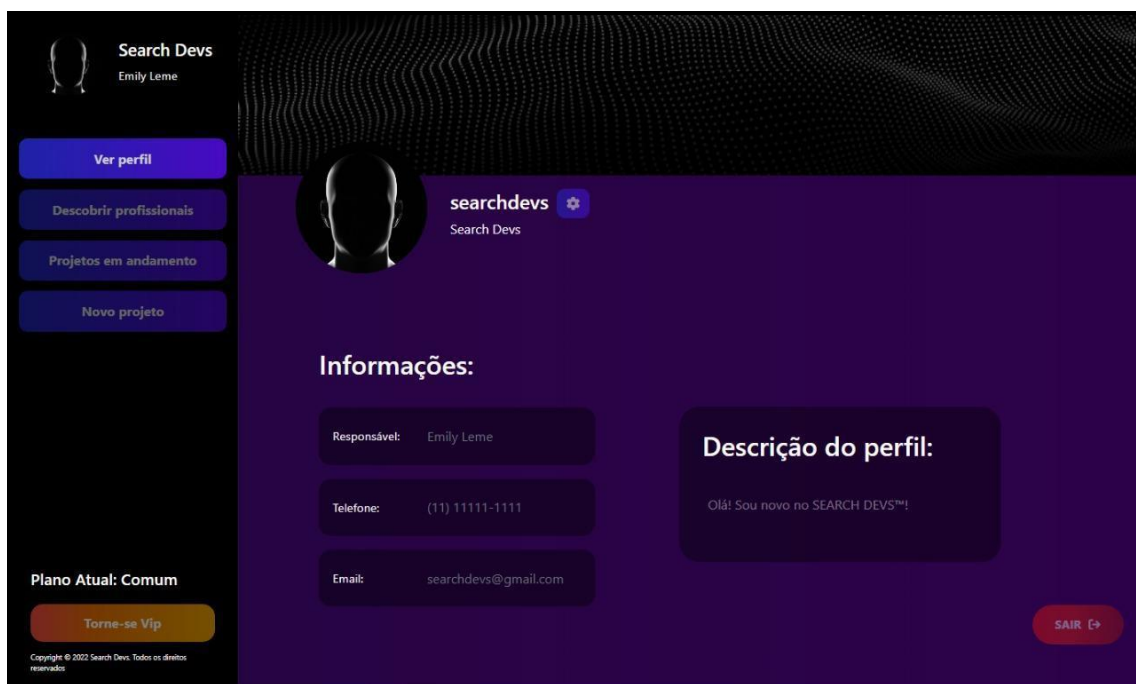
Figura 52 - Perfil do Desenvolvedor



Fonte: Autoria Própria, 2022.

O perfil do empreendedor pode ser visualizado na Figura 53, nela são mostradas as informações da empresa como o responsável e meios de contato como o número de telefone, também é possível ler a descrição que foi deixada pela empresa.

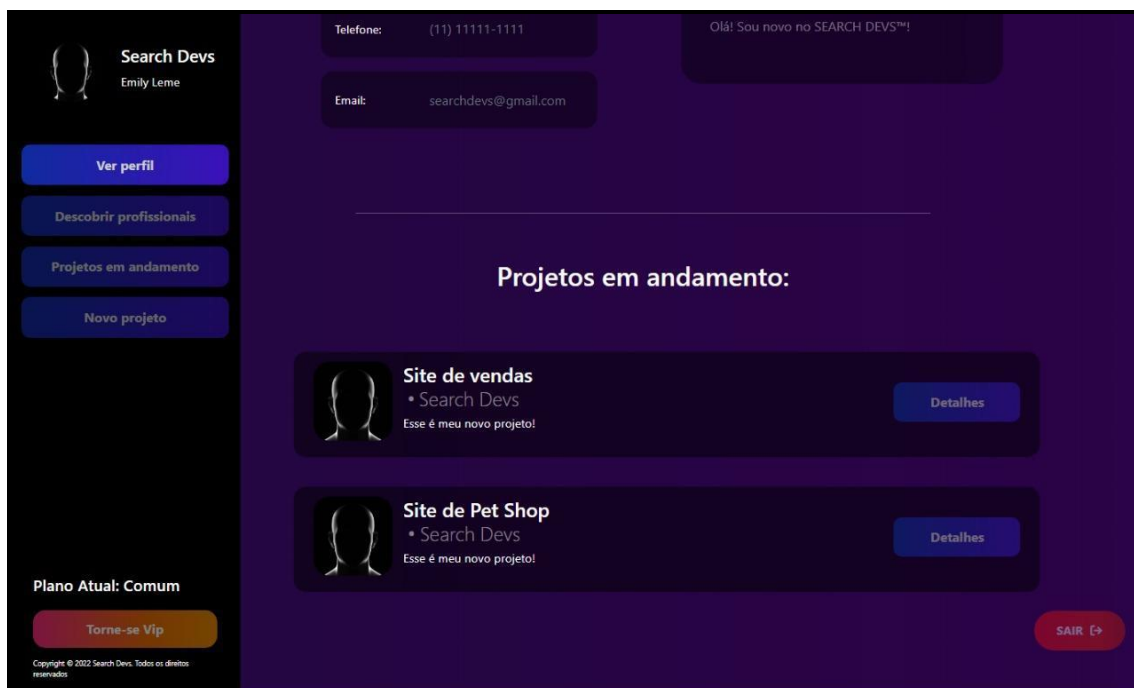
Figura 53 - Perfil Contratante



Fonte: Autoria Própria, 2022.

Descendo um pouco mais a página, é possível visualizar a Figura 54 com alguns dos projetos que foram criados pela empresa e que ainda não foram concluídos, com botões para acessar o projeto se quiser saber mais detalhes.

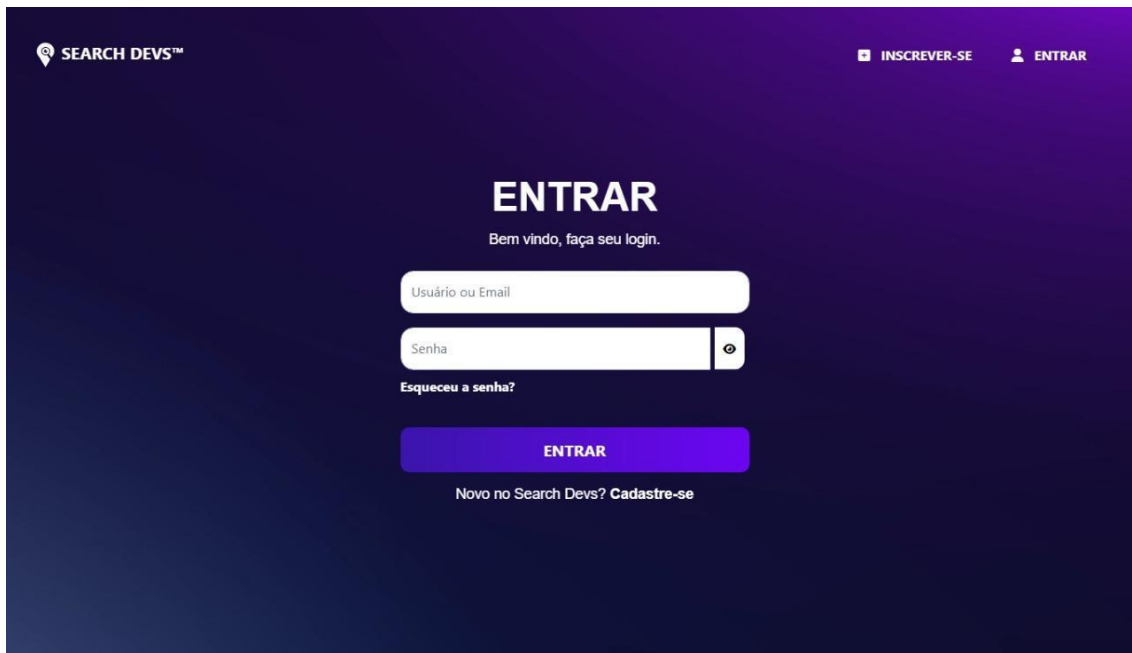
Figura 54 - Perfil Contratante 2



Fonte: Autoria Própria, 2022.

A Figura 55 mostra a tela de login. O usuário deverá preencher o formulário de acordo com o que foi cadastrado antes para que não ocorra problemas na hora do acesso a sua própria conta. É necessário que o usuário espere a validação de seus dados para continuar a utilizar o site.

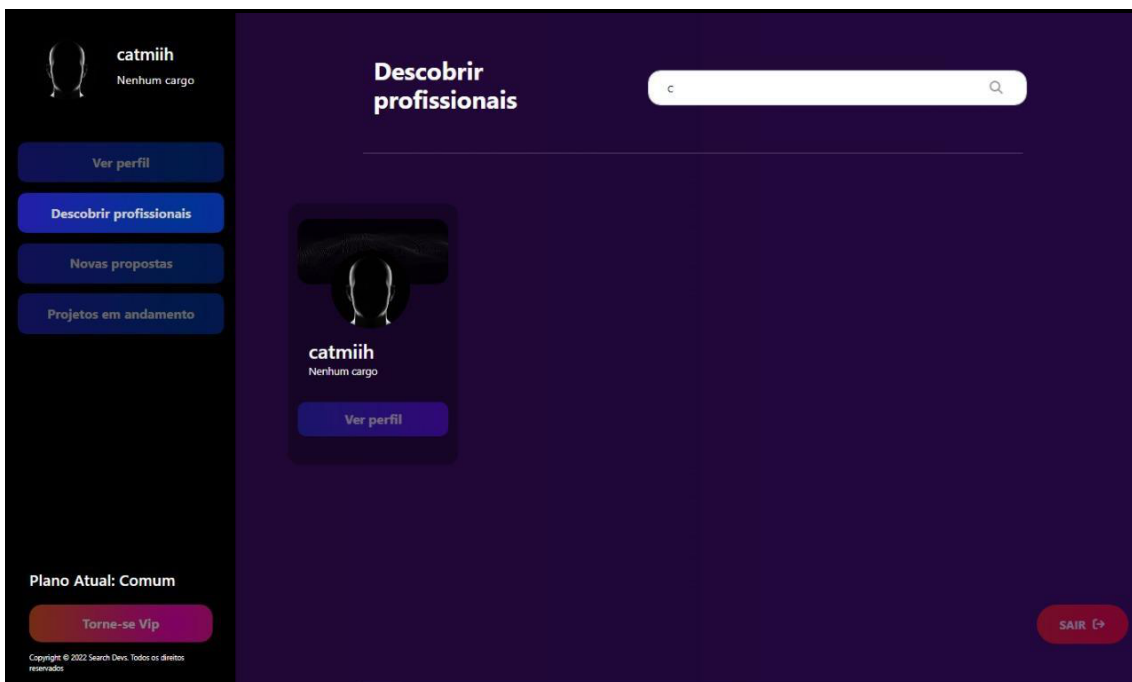
Figura 55 - Tela de Login



Fonte: Autoria Própria, 2022.

A Figura 56 apresenta a parte de pesquisa tanto para o desenvolvedor como para o contratante. O usuário pode acessar a aba descobrir profissionais e digitar o nome de usuário do perfil que deseja encontrar.

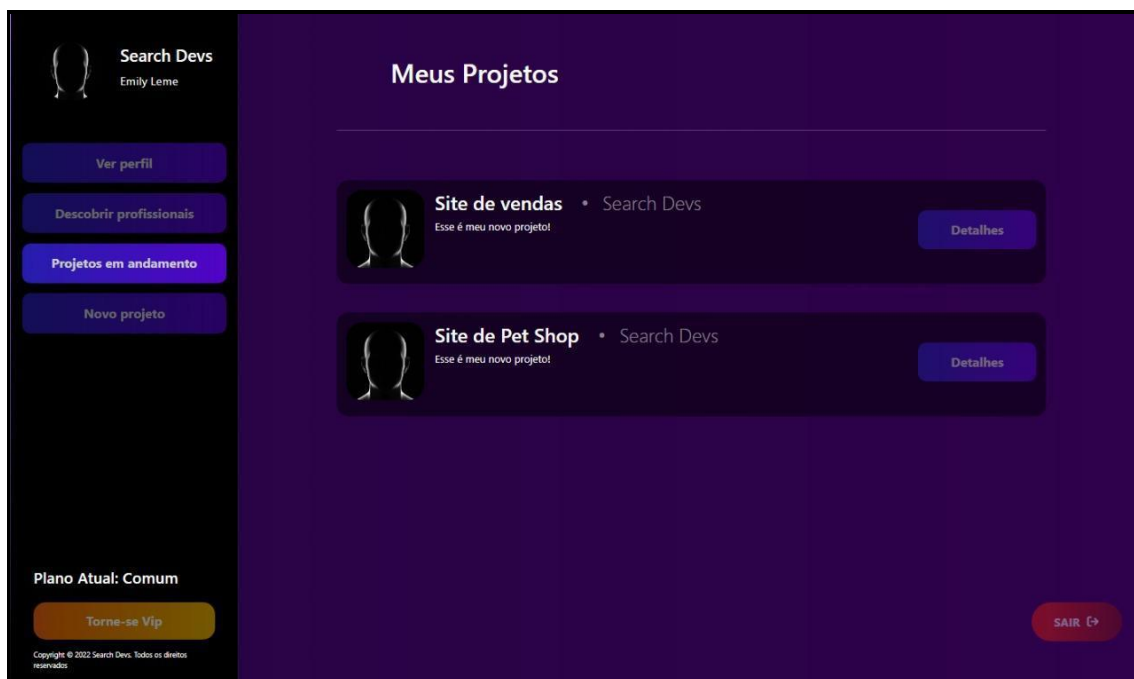
Figura 56 - Pesquisar Perfis



Fonte: Autoria Própria, 2022.

Na aba projetos em andamento demonstrado na Figura 57, o empreendedor consegue visualizar todos os projetos que ele criou e que está em desenvolvimento.

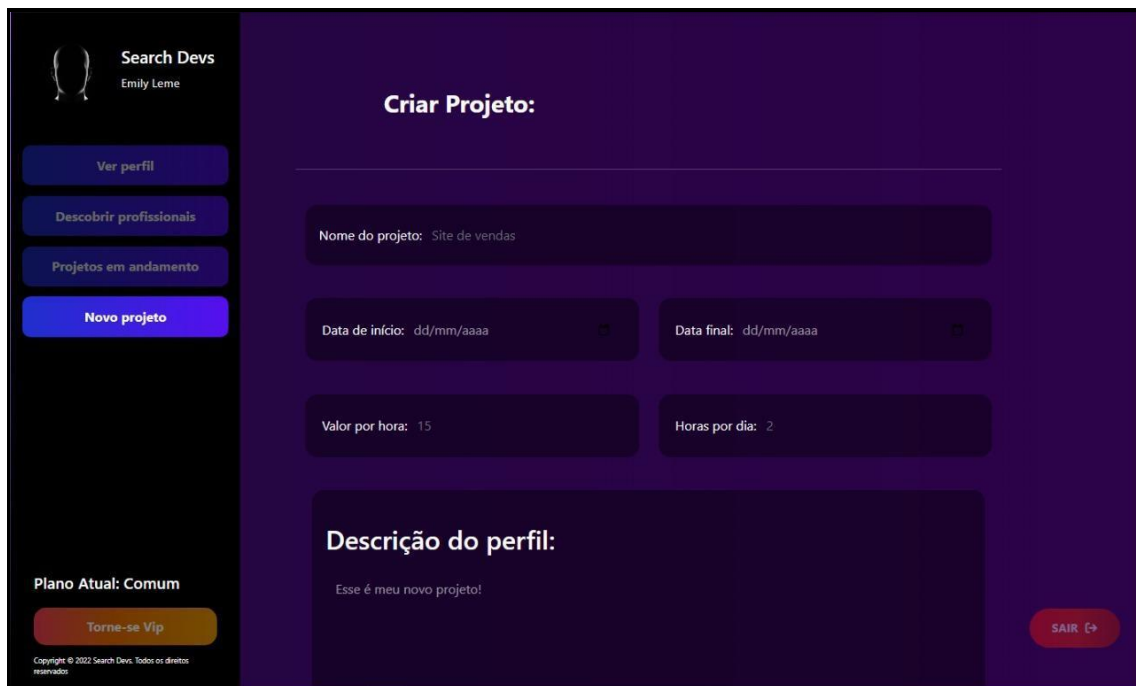
Figura 57 - Projetos em Andamento



Fonte: Autoria Própria, 2022.

Ao clicar no botão de Novo Projeto a empreendedor é direcionado para a página de criação de novo projeto representada na figura 58. Ao preencher o formulário inserindo data de início e conclusão, o valor, as horas por dia, o nome e a descrição do seu projeto, são enviados ao banco de dados e a A.I faz a busca pelo desenvolvedor ideal.

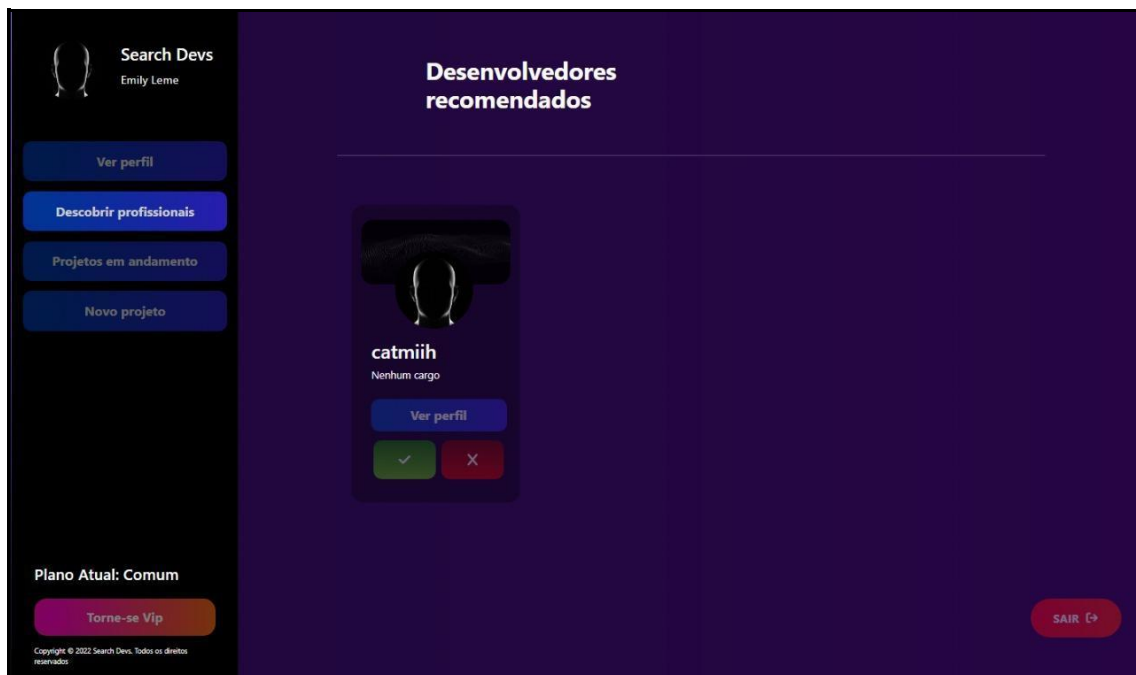
Figura 58 - Criação de Projeto



Fonte: Autoria Própria, 2022.

Após um tempo a A.I envia os desenvolvedores mais adequados registrados no Search Devs para que o empreendedor visualize o perfil e decida se irá aceitá-lo no projeto, como mostrado na figura 59.

Figura 59 - Desenvolvedores recomendados

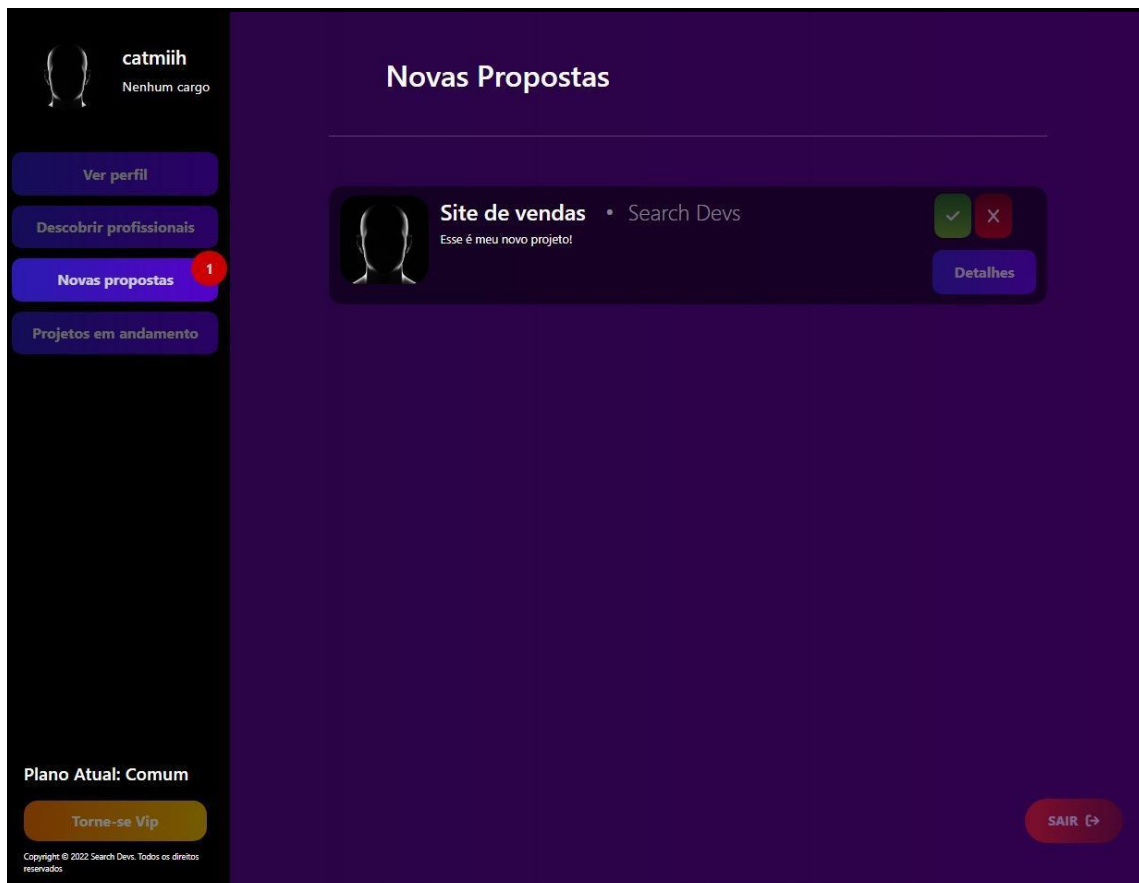


Fonte: Autoria Própria, 2022.

Na figura 60 temos a tela de solicitação ao desenvolvedor. Se aceito pelo empreendedor, é enviada a proposta para participar do projeto à o desenvolvedor e ele pode aceitar a solicitação ou não.

Se aceito, é enviada a solicitação para participar do projeto para o desenvolvedor.

Figura 60 - Novas Propostas



Fonte: Autoria Própria, 2022.

4 CONCLUSÃO

As pesquisas feitas inicialmente a respeito da contratação no mercado de tecnologia, nortearam o projeto do início ao fim. Ao tentar encontrar formas de auxiliar a contratação fomos além, não apenas facilitando a contratação, mas também na busca pelo serviço ideal para cada projeto.

Através dos diagramas UML foi possível fazer planejamentos a longo prazo dentro da aplicação para prevenção de falhas. Os diagramas de classe, sequência, atividade e caso de uso auxiliaram na visualização do projeto de outras perspectivas fazendo com que fosse possível deixar a aplicação mais intuitiva e acessível.

O PHP foi de grande valia no projeto, pois através dele foi feita a conexão com o banco de dados, a criação das classes e das respectivas telas. A criação da A.I foi desenvolvida também em PHP para facilitar a implementação.

Alcançando o propósito, a aplicação foi submetida a testes com desenvolvedores e empreendedores e mais uma vez passada pela fase de melhorias, novas implementações. Juntamente com ele, nos utilizamos das tecnologias HTML5, CSS3, JavaScript, Bootstrap e JQuery, o que tornou o projeto o mesmo no qual planejamos.

Com ele, conseguimos alcançar todos os nossos objetivos como equipe, que foi entregar uma plataforma acessível para todos, prática, e parcialmente automatizada, assim tornando-a uma plataforma acessível, simples e útil para todos.

REFERÊNCIAS

ACHOUR, Mehdi et al. **PHP: manual do php. Manual do PHP.**2022. The PHP Documentation Group. Disponível em: https://www.php.net/manual/pt_BR/. Acesso em: 09 ago. 2022.

ALVES, William Pereira. **Banco de Dados.** São Paulo: Editora Érica Ltda., 2013. 160 p.

ARAÚJO, João Paulo. **COM GRANDE DIFICULDADE PARA ATRAIR TALENTOS DE TECNOLOGIA, EMPRESAS COMEÇAM A CONTRATAR FREELANCERS.** Disponível em: https://www.gestaoerh.com.br/pt_br/artigos/8304-com-grande-dificuldade-para-atrair-talentos-de-tecnologia-empresas-comecam-a-contratar-freelancers. Acesso em: 26 abr. 2022.

BARBOSA, Gabriel. **Conhecendo a biblioteca jQuery.** 2011. Disponível em: <https://www.devmedia.com.br/conhecendo-a-biblioteca-jquery/22778>. Acesso em: 13 jun. 2022.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML: guia do usuário.** 2. Ed. Rio de Janeiro: Elsevier Editora Ltda., 2006. 552 p.

CONVERSE, Tim; PARK, Joyce. **PHP: a bíblia.** 2. ed. Rio de Janeiro: Elsevier Editora Ltda., 2003. Acesso em: 09 ago. 2022.

DATE, C. J. **Introdução a Sistemas de banco de dados.** 8. ed. Rio de Janeiro: Elsevier Editora Ltda., 2004. 896 p.

EIS, Diego; FERREIRA, Elcio. **HTML5 e CSS3 com farinha e pimenta**. São Paulo: Clube de Autores, 2012. 219 p. Acesso em: 8 ago.2022.

FERREIRA, Silvio. **Guia prático de HTML5**. São Paulo: Universo dos Livros, 2013. 135 p. Acesso em: 8 de ago. 2022.

FOWLER, Martin. **UML Essencial: Um Breve Guia para Linguagem Padrão**. [S. l.]: Bookman Editora, 2014. 162 p. ISBN 9788560031382.

LARMAN, Craig. **Utilizando UML e Padrões**: uma introdução a análise e ao projeto orientado a objetos e ao desenvolvimento iterativo. 3. Ed. Porto Alegre: Bookman Companhia Editora, 2007. 696 p.

LIMA, Adilson da Silva. **UML 2.0**: do requisito à solução. 4. ed. São Paulo: Editora Érica Ltda., 2010. 326 p.

Lima, Guilherme. “**Bootstrap: O Que é, Como Instalar E Exemplos Práticos.**” **Alura**, 21 jul. 2021. Disponível em: www.alura.com.br/artigos/bootstrap. Acesso em: 8 ago. 2022.

MILANI, André. **MySQL: Guia do Programador**. São Paulo: Novatec, 2007. 400 p. ISBN 8575221035.

MORAZ, Eduardo. **Treinamento Prático em PHP**. Sp: Digerati Books, 2005. 181 p. Acesso em: 09 ago. 2022.

MOZILLA. **CSS** | **MDN**. 26 jun. 2021a. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/CSS>. Acesso em: 12 jun. 2022.

MOZILLA. **HTML: Linguagem de Marcação de Hipertexto | MDN**. 7 nov. 2021b. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/HTML>. Acesso em: 8 jun. 2022.

MOZILLA. **JavaScript básico - Aprendendo desenvolvimento web | MDN**. 19 ago. 2021d. Disponível em: https://developer.mozilla.org/pt-BR/docs/Learn/Getting_started_with_the_web/JavaScript_basics. Acesso em: 13 jun. 2022.

MOZILLA. **JavaScript | MDN**. 2 ago. 2021c. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript>. Acesso em: 12 jun. 2022.

MOZILLA. **jQuery - Glossário | MDN**. 11 fev. 2021d. Disponível em: <https://developer.mozilla.org/pt-BR/docs/Glossary/jquery>. Acesso em: 13 jun. 2022.

MYSQL AB (org.). **Manual de Referência do MySQL 4.1**. 4.1 [S. L.]: Equipe de Documentação da Mysql, 2010. 861 p. Disponível em: <https://downloads.mysql.com/docs/refman-4.1-pt.a4.pdf>. Acesso em: 13 jul. 2022.

SILVA, Maurício. **CSS3: desenvolva aplicações web profissionais com o uso dos poderosos recursos de estilização das css3**. São Paulo: Novatec Editora Ltda., 2012. 496 p. Acesso em: 8 de ago. 2022.

SILVA, Maurício. **Fundamentos de HTML5 e CSS3**. São Paulo: Novatec Editora Ltda., 2015. 304 p. Acesso em: 8 de ago. 2022.

SILVA, Maurício. **JavaScript - Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript**. [S. l.]: Novatec Editora, 2020. Acesso em: 12 jun. 2022.

SILVA, Maurício. **jQuery. A Biblioteca do Programador Javascript**. 3. ed. [S. l.]: Novatec, 2013. E-book (544 p.). ISBN 8575223879

STADZISZ, Prof. Paulo César. **Projeto de Software usando a UML**. 2002. 69 f. Dissertação (Mestrado) - Curso de Informática, Departamento Acadêmico de Informática, Centro Federal de Educação Tecnológica do Paraná, Paraná, 2002.

T. A. GUEDES, Gilleanes. **UML 2 - Uma Abordagem Prática**. 3. ed. São Paulo: Novatec Editora, 2018. 488 p. ISBN 8575222813.

RAMAKRISHNAN, Raghu; GEHRKE, Johannes. **Sistemas de Gerenciamento de Banco de Dados**. 3. ed. São Paulo: McGraw-Hill Interamericana do Brasil Ltda., 2007. 905 p.

SOMMERVILLE, Ian. **Engenharia de Software**. 9. ed. São Paulo: Pearson Universidades, 2013. 529 p.