**CENTRO ESTADUAL DE EDUCAÇÃO TECNOLÓGICA PAULA SOUZA**

**ETEC DA ZONA LESTE**

**NOVOTEC DESENVOLVIMENTO DE SISTEMAS**

**Baneza Livia Coarite Manzaneda**

**Diogo Marques Moreira**

**Henrique de Moraes Rodrigues**

**João Pedro da Silva Machado Felix**

**NUEVOS CAMINOS:**

**Sistema auxiliador e informativo aos imigrantes em empregabilidade e moradia**

**São Paulo**

**2024**

**Baneza Livia Coarite Manzaneda**

**Diogo Marques Moreira**

**Henrique de Moraes Rodrigues**

**João Pedro da Silva Machado Felix**

**NUEVOS CAMINOS: Sistema auxiliador e informativo aos imigrantes em empregabilidade e moradia**

Trabalho de Conclusão de Curso apresentado ao Curso do Ensino Médio com Habilitação Profissional de Técnico em Desenvolvimento de Sistemas da Etec Zona Leste, orientado pelo Prof. Jeferson Roberto de Lima, como requisito parcial para a obtenção do título de técnico em Desenvolvimento de Sistemas.

**São Paulo**

**2024**

**RESUMO**

Nuevos Caminos visa auxiliar e informar sobre a empregabilidade e moradia aos imigrantes que pretendem vir ao Brasil. O foco do projeto é melhorar a compreensão do território brasileiro, contendo informações da empregabilidade e moradia, além de proporcionar um fórum interativo entre imigrantes, assim possibilitando tirar suas dúvidas, trocar informações e experiências vivenciadas. O desenvolvimento do projeto tem como propósito amenizar as causas de desafios e conflitos enfrentados pelos imigrantes. Com base na aprendizagem, serão utilizadas medidas de auxiliar e compreender os imigrantes que vêm ao território brasileiro, analisando possibilidades e situações que possam gerar suas dúvidas e preocupações, para podermos auxiliar esse imigrante em tal cenário, dúvida, preocupação ou outros conflitos que o deixam desorientados no novo ciclo de sua vida.

**Palavras-chave:** Imigrantes. Brasil. Empregabilidade. Moradia. Auxiliar.

**ABSTRACT**

First, Nuevo’s Caminos aims to assist and inform about employability and housing for immigrants who intend to come to Brazil. The focus of the project is to improve the understanding of Brazilian territory, containing information on employability and housing, in addition to providing an interactive forum for immigrants, thus making it possible to answer their questions and exchange information and experiences. The project's development aims to alleviate the causes of challenges and conflicts faced by immigrants. Based on learning, measures will be used to assist and understand immigrants who come to Brazilian territory, analyzing possibilities and situations that may generate their doubts and concerns so that we can help these immigrants in such scenarios, doubt, problems, or other conflicts that leave them disoriented in the new cycle of their life.

**Keywords:** Immigrants. Brazil. Employability. Housing. Assist.

**LISTA DE ILUSTRAÇÕES**

[Figura 1 - Exemplo HTML 12](#_Toc181368347)

[Figura 2 - Resultado exemplo HTML 13](#_Toc181368348)

[Figura 3 - Exemplo relação do HTML com o CSS 14](#_Toc181368349)

[Figura 4 - Código de um formulário simples em CSS 15](#_Toc181368350)

[Figura 5 - Resultado da página web com o CSS 17](#_Toc181368351)

[Figura 6 - Exemplo JavaScript 18](#_Toc181368352)

[Figura 7 - Resultado JavaScript na página 19](#_Toc181368353)

[Figura 8 - Exemplo DER 22](#_Toc181368354)

[Figura 9 - Exemplo MER 23](#_Toc181368355)

[Figura 10 - Exemplo MySQL 26](#_Toc181368356)

[Figura 11 - Visualização MySQL 27](#_Toc181368357)

[Figura 12 - Exemplo código React Native 33](#_Toc181368358)

[Figura 13 - Exemplo finalizado React Native 36](#_Toc181368359)

[Figura 14 - Exemplo Wireframe baixa fidelidade 37](#_Toc181368360)

[Figura 15 - Exemplo Wireframe de alta fidelidade 37](#_Toc181368361)

[Figura 16 - Exemplo diagrama Caso de uso 39](#_Toc181368362)

[Figura 17 - Exemplo diagrama de atividade 40](#_Toc181368363)

[Figura 18 - Exemplo diagrama de sequência 41](#_Toc181368364)

[Figura 19 - Exemplo diagrama de classes 42](#_Toc181368365)

[Figura 20 - Diagrama de Caso de Uso do Sistema 44](#_Toc181368366)

[Figura 21 - Diagramade Atividade: Cadastrar Imigrante 52](#_Toc181368367)

[Figura 22 - Diagrama de Atividade: Fazer login 53](#_Toc181368368)

[Figura 23- Diagrama de Atividade: Interagir com outras postagens 54](#_Toc181368369)

[Figura 24 - Diagrama de Atividade: Manter informações 55](#_Toc181368370)

[Figura 25 - Diagrama de Atividade: Manter Postagem do Imigrante 56](#_Toc181368371)

[Figura 26 - Diagrama de Atividade: Manter Postagens 57](#_Toc181368372)

[Figura 27 - Diagrama de Atividade: Pesquisar Informações 58](#_Toc181368373)

[Figura 28 - Diagrama de Atividade: Visualizar Fórum 59](#_Toc181368374)

[Figura 29 - Diagrama de Atividade: Visualizar Informações 60](#_Toc181368375)

[Figura 30 – Diagrama de Sequência: Cadastrar Imigrante 62](#_Toc181368376)

[Figura 31 - Diagrama de Sequência: Fazer Login 63](#_Toc181368377)

[Figura 32 - Diagrama de Sequência: Interagir com outras postagens 64](#_Toc181368378)

[Figura 33 - Diagrama de Sequência: Manter Informações 65](#_Toc181368379)

[Figura 34 - Diagrama de Sequência: Manter Postagem do Imigrante 66](#_Toc181368380)

[Figura 35 - Diagrama de Sequência: Manter Postagens 67](#_Toc181368381)

[Figura 36 - Diagrama de Sequência: Pesquisar Informação 68](#_Toc181368382)

[Figura 37 - Diagrama de Sequência: Visualizar Fórum 69](#_Toc181368383)

[Figura 38 - Diagrama de Sequência: Visualizar Informações 70](#_Toc181368384)

LISTA DE QUADROS

[Quadro 1 – Descrição do caso de uso: Fazer Login 45](#_Toc181361374)

[Quadro 2 – Descrição do casos de uso: Cadastrar Imigrante 46](#_Toc181361375)

[Quadro 3 – Descrição do casos de uso: Visualizar Informação 47](#_Toc181361376)

[Quadro 4 - Documentação do caso de uso: Pesquisar Informações 47](#_Toc181361377)

[Quadro 5 - Descrição do caso de uso: Visualizar Fórum 48](#_Toc181361378)

[Quadro 6 - Descrição do caso de uso: Manter a postagem do imigrante Fórum 49](#_Toc181361379)

[Quadro 7 – Documentação do caso de uso: Interagir com outras postagens 49](#_Toc181361380)

[Quadro 8 - Documentação do caso do uso: Manter Informação 50](#_Toc181361381)

[Quadro 9 - Documentação de caso de uso: Manter postagem 50](#_Toc181361382)

**LISTA DE ABREVIATURAS E SIGLAS**

Cascading Style Sheet (CSS)

Diagrama de Entidade Relacionamento (DER)

Modelo Entidade-Relacionamento (MER)

PHP: Hypertext Preprocessor (PHP)

Structured Query Language (SQL)

Node Package Manager (NPM)

Hypertext Markup Language (HTML)

JavaScript (JS)

Sistema de gerenciador de banco de dados (SGBD)

Unified Modeling Language (UML)

Sumário

[1. INTRODUÇÃO 9](#_Toc181361423)

[2. REFERENCIAL TEÓRICO 10](#_Toc181361424)

[**2.1. Imigrantes dentro do território brasileiro 10**](#_Toc181361425)

[**2.2. Linguagens Utilizadas 10**](#_Toc181361426)

[**2.2.1. Html 11**](#_Toc181361427)

[**2.2.2. CSS 13**](#_Toc181361428)

[**2.2.3. JavaScript 17**](#_Toc181361429)

[**2.2.4. SQL 19**](#_Toc181361430)

[**2.2.5. Banco de dados relacional 20**](#_Toc181361431)

[**2.2.6. Der 21**](#_Toc181361432)

[**2.2.7. Mer 22**](#_Toc181361433)

[**2.2.8. PHP 23**](#_Toc181361434)

[**2.2.8. Mysql 25**](#_Toc181361435)

[**2.2.9. Node.js 27**](#_Toc181361436)

[**2.2.10. Npm 28**](#_Toc181361437)

[**2.2.11. Expo 29**](#_Toc181361438)

[**2.2.12. React 30**](#_Toc181361439)

[**2.2.13. Bootstrap 31**](#_Toc181361440)

[**2.2.14. React Native 32**](#_Toc181361441)

[**2.2.15. Wireframes 36**](#_Toc181361442)

[**2.2.16. Unified Modeling Language 37**](#_Toc181361443)

[**2.2.18. Diagrama de caso de uso 38**](#_Toc181361444)

[**2.2.17. Diagrama de Atividades 39**](#_Toc181361445)

[**2.2.18. Diagrama de sequência 40**](#_Toc181361446)

[**2.2.19. Diagrama de classes 41**](#_Toc181361447)

[3. Desenvolvimento 43](#_Toc181361448)

[3.1. Diagrama de Caso de Uso 43](#_Toc181361449)

[3.2. Documentação dos Casos de Usos 44](#_Toc181361450)

[3.3. Diagrama de Atividade 51](#_Toc181361451)

[3.4. Diagrama de Sequência 61](#_Toc181361452)

[REFERÊNCIAS 79](#_Toc181361453)

# 1. INTRODUÇÃO

*Nuevos Caminos* aborda o desenvolvimento de um sistema informativo e de apoio que visa facilitar a empregabilidade e o acesso a moradias para imigrantes que chegam ao Brasil. Esse projeto tem como extrema relevância na importância dos imigrantes que pretendem adentrar ao Brasil, com a intenção de combater as falsas notícias sobre o Brasil.

De acordo com Migalhas, a advogada especializada em imigração Mara Pessoni afirma que os imigrantes têm diversos problemas de habitação no Brasil e falta de oportunidades de emprego, porque eles são um grupo em estado de maior vulnerabilidade no Brasil. (MIGALHAS, 2023).

Uma pesquisa realizada pela ONG Visão Mundial revelou que uma grande parcela dos imigrantes está fora do mercado de trabalho, representando 67,4%. Entre 2011 e 2021, inúmeros imigrantes no mercado de trabalho no Brasil eram de 62 mil desempregados. Atualmente, o Brasil tem por volta de 1,5 milhão de imigrantes desempregados (Rodrigues, 2024). Para Bepay Bank, os imigrantes vêm ao território brasileiro com o principal objetivo de melhorar sua condição vida, renda, oportunidades de empregos e estudos (G1, 2024).

Com essas informações, a realização do projeto, os imigrantes terão uma nova visão quando vierem ao território nacional brasileiro, por meio de informações de como ter um recomeço próspero e eficaz para a sua estadia. Contendo ensinamentos desde como funciona um currículo brasileiro até como é cada estado brasileiro e suas regiões, informações sobre o Índice de Desenvolvimento Humano Municipal (IDHM), de acordo com os estados brasileiros.

Os imigrantes enfrentam diversos problemas quando chegam ao Brasil e a maioria deles é: dificuldade de entrar no mercado de trabalho e dificuldade de encontrar um lar.

Então como o nosso sistema pode auxiliar e informar os imigrantes dentro do mercado de trabalho e sua moradia enquanto estiverem no Brasil?

O projeto tem como objetivo um website e um app informativo a imigrantes que desejam entrar no Brasil, melhorando sua compreensão sobre o território nacional, contendo informações que abranjam a moradia e empregabilidade, além do fórum interativo entre os usuários para que possam esclarecer alguma dúvida e troquem informações e experiências vivenciadas entre si.

Em nosso próximo capítulo, apresentaremos diversos livros de autores especializados em programação como base para orientarmos no desenvolvimento do projeto. Assim, destacaremos as principais linguagens de programação que iremos utilizar para desenvolvermos o projeto *Nuevos Caminos*, entre elas são: HTML, CSS, JavaScript, PHP, Node.js, entre outras.

# 2. REFERENCIAL TEÓRICO

O capítulo tem como objetivo mostrar toda a teoria que foi utilizada no trabalho, colocando toda tecnologia utilizada e os tópicos principais para a conclusão do projeto.

## **2.1. Imigrantes dentro do território brasileiro**

De acordo com Migalhas, a advogada especializada em imigração Mara Pessoni afirma que os imigrantes têm diversos problemas de habitação no Brasil e falta de oportunidades de emprego, porque eles são um grupo em estado de maior vulnerabilidade no Brasil. (MIGALHAS, 2023).

Conforme Bepay Bank os Imigrantes têm como objetivo principal, na sua grande maioria, a melhora em sua condição de vida, conseguir uma renda com oportunidades de emprego além de estudos (G1, 2024).

## **2.2. Linguagens Utilizadas**

Segue abaixo as linguagens utilizadas que estão Incluídas dentro do projeto web e aplicação móvel para uma melhor experiência do usuário.

### **2.2.1. Html**

De acordo com Flatschart (2011), a sigla HTML significa *HyperText Markup* *Language* (Linguagem de Marcação de Hipertexto) e é a principal linguagem utilizada no desenvolvimento web.

O HTML é uma linguagem de marcação, utilizada para criar páginas de sites com textos e informações. Estas páginas podem ser acessadas por qualquer pessoa e por qualquer tipo de navegador, afirma Carril (2012).

Segundo Quierelli (2012), a estrutura de um documento HTML consiste em comandos, que também podem ser chamados de *tags*, que devem ser utilizadas na maioria das vezes em pares.

Figura 1 - Exemplo HTML

Texto

Descrição gerada automaticamente

Fonte: Do Próprio Autor, 2024.

* *<DOCTYPE>:* Diz qual versão do HTML está sendo usada nesse código.
* *<html>*: É a *tag* principal do código, ela que engloba todas as outras *tags*.
* *<head>*: É responsável por armazenar as configurações do código.
* *<link>*: Especifica uma ou mais relações deste documento atual a um recurso externo.
* *<title>*: Diz qual é o título da página que será aberta no browser.
* *<body>*: Armazena todo o conteúdo da página que será mostrado no *browser*.
* *<h1>*: Representa o nível de título mais alto, sendo de 1 a 6, também conhecida como tag de cabeçalho.
* *<form>*: Indica ao sistema que estamos iniciando um formulário, sendo possível receber diversos métodos e ações.
* *<label>*: Representa uma legenda dentro da interface do usuário.
* *<input>*: É um recurso interativo onde o usuário pode inserir alguma coisa.
* *<button>*: Cria um botão ao usuário, sendo possível executar alguma ação.
* *<div>*: Defina uma seção no documento, facilitando a estilização dessas áreas.
* *<script>*: Cria um espaço para que seja possível incorporar o código do javascript.

Figura 2 - Resultado exemplo HTML

Uma imagem contendo Diagrama

Descrição gerada automaticamente

Fonte: Do Próprio Autor, 2024.

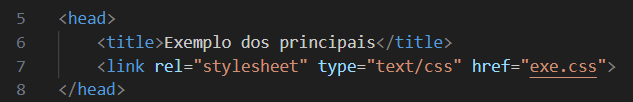
### **2.2.2. CSS**

O Cascading Style Sheets (CSS), é uma linguagem que, segundo Albino (2023), ajuda a descrever a forma e a aparência dos elementos em HTML, como definir o estilo da página, cores, fontes, além de oferecer um melhor design para o site.

Essa linguagem permite que o desenvolvedor possa criar normas, para conduzir como um elemento do HTML possa ser visualizado, permitindo a alteração das caixas desses elementos, como afirma Duckett (2016).

De acordo com Zemel (2015), há seletores no CSS que podem facilitar a maneira de codificar o design das páginas, para que o interpretador do navegador possa encontrar as instâncias, além do ID, classe e o tipo escrito no arquivo.

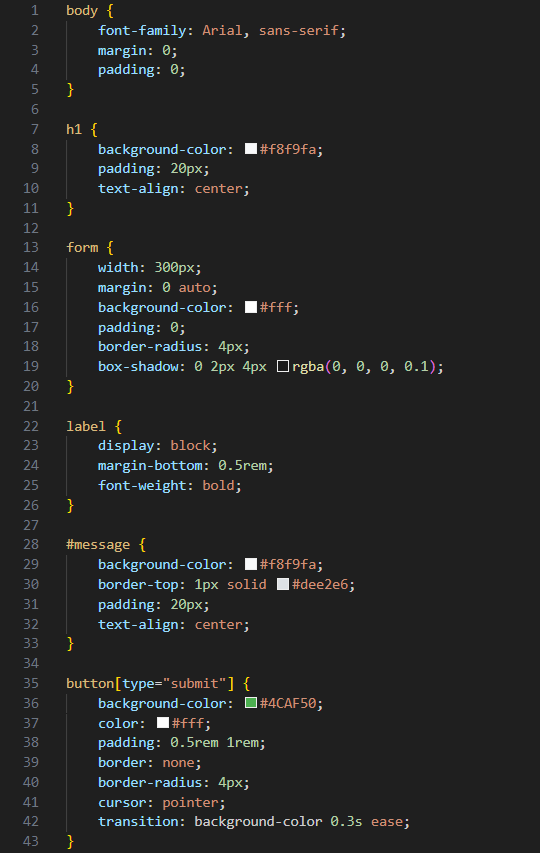
Figura 3 - Exemplo relação do HTML com o CSS



Fonte: Do próprio autor, 2024.

Em seguida, estará um exemplo completo do código CSS de um formulário simples.

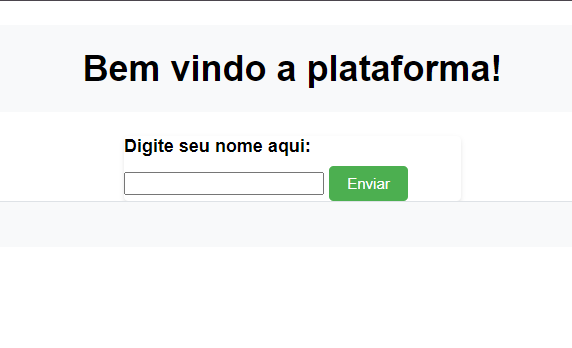
Figura 4 - Código de um formulário simples em CSS



Fonte: Do próprio autor, 2024.

* *body*: Uma classe de manipulação de elementos dentro do corpo do HTML.
* *font-family*: Definição da fonte usada
* *margin*: Definição da margem usada nos quatro lados da aplicação
* *padding*: É o espaço entre a borda de um elemento e seu conteúdo.
* *background-color*: Define a cor de fundo de um elemento.
* *text-align*: O alinhamento do texto.
* *width*: Define a largura da área do conteúdo.
* *boder-radius*: Arredonda as bordas de um único elemento.
* *box-shadow*: Utilizada para adicionar efeitos de sombra entorno de um elemento.
* *display*: Define o tipo de exibição interno e externo de um elemento.
* *margin-bottom*: define o tamanho da área abaixo de um elemento.
* *fonte-weight*: Utilizada para definir a intensidade da luz ou peso da fonte.
* *border-top*: Define todas as propriedades da borda superior de um elemento.
* *border*: Define a borda de um elemento.
* *cursor*: determina o cursor do mouse quando houver interação.
* *transition*: Define a trançam de dois estados de um elemento.

Figura 5 - Resultado da página web com o CSS



Fonte: Do próprio autor, 2024.

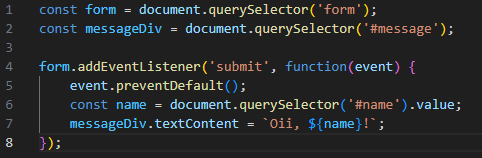
### **2.2.3. JavaScript**

A linguagem JavaScript é utilizada mundialmente, sendo um dos 3 pilares do desenvolvimento web.

Segundo Flanagan (2012), Javascript é uma linguagem de programação web que as maiores parcelas de sites a usam, por ser uma linguagem de ponta, bem estruturada e completa auxiliando outras linguagens orientadas a objetos.

Porém, não podemos nos dar ao luxo de fazer o que quisermos, porque nesta linguagem tudo tem seu local de trabalho e principalmente é um especialista em armazenar, de acordo com Morrison (2020).

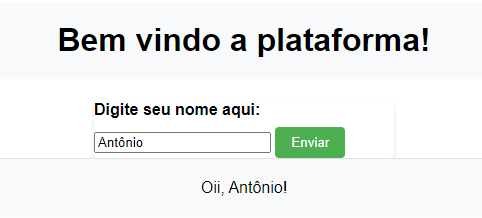
Figura 6 - Exemplo JavaScript



Fonte: Do Próprio Autor, 2024.

* *const*: cria uma variável que seja continua e não alterável, sendo somente leitura da variável.
* *document.querySelector*: é um método dentro do JavaScript que permite selecionar um único elemento dentro do HTML.
* *addEventListener*: Um manipulador de eventos, o que permite realizá-los.
* *preventDefault*: Faz que um evento se cancele, quando possível.
* *value*: Serve para se obter um valor dentro de um campo.
* *textContent*: retorna todo conteúdo utilizados em textos, incluindo o css utilizado.

Figura 7 - Resultado JavaScript na página



Fonte: Do Próprio Autor, 2024.

### **2.2.4. SQL**

Conforme destacado por Prescott (2015) o *Structured Query Language* (SQL) é uma linguagem de consulta que irá conversar com bancos de dados relacionais. O SQL é utilizado para inserir, recuperar, atualizar e excluir dados em um banco de dados.

Segundo Microsoft o SQL é uma linguagem de consulta mais utilizada no mundo da tecnologia do banco de dados de código aberto por conta da sua flexibilidade, facilidade de utilizar e sua variação de interação com outras linguagens de programação.

### **2.2.5. Banco de dados relacional**

Banco de dados relacional é um banco que fornece acesso a dados relacionados entre si. Os bancos de dados relacionais oferecem uma abordagem simples e direta para representar dados em tabelas. (ORACLE).

O banco de dados relacional pode facilitar a organização de inúmeros dados estruturados em uma tabela criada pela sua conexão a meio de linhas e colunas que podem ser ligadas por uma chave primária ou estrangeira. (IBM).

Em todas as tabelas do banco de dados relacional possuem atributos conhecidos como chave primária, ela é um identificador de linha que não se repete. Com tabelas diferentes usa-se a chave estrangeira que se refere a chave primária. (Cloud Google).

### **2.2.6. DER**

De acordo com Bazzi (2013) o diagrama entidade relacionamento (DER) tem como o principal objetivo identificar as entidades nos dados e a forma de como estão relacionadas entre elas.

Figura 8 - Exemplo DER

Diagrama

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Este exemplo mostra um simples DER de um agendamento de consulta médica, indicando que um paciente tem várias consultas (1 para N), juntamente com o médico.

### **2.2.7. MER**

Segundo Elmasri (2005), o modelo entidade relacionamento (MER) é um modelo de dados conceitual de alto nível e sendo muito popular, sendo utilizado normalmente a projetos que contenha o banco de dados.

Figura 9 - Exemplo MER

Diagrama, Esquemático

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Este exemplo compõe o anterior do DER (Figura 8), vemos que as entidades tiveram suas especificações para se tornarem únicas para não ter conflitos com outras entidades do mesmo tipo, sendo principalmente diferenciada pelo ID que é sua *Primary key* (PK) onde cada dado guardado se torna único.

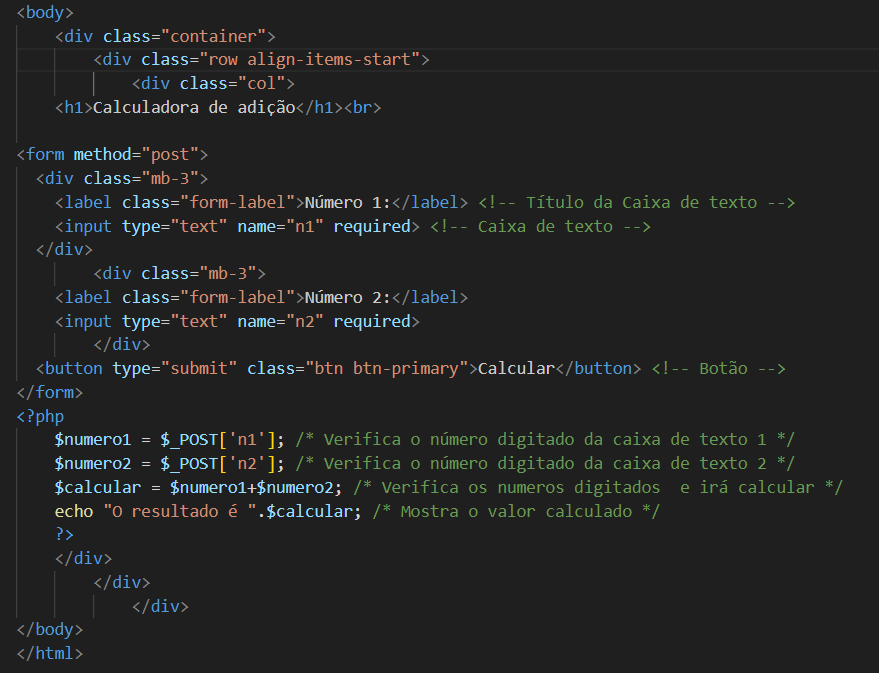
Passando o ID do paciente e Médico para a consulta onde eles se tornam uma *Foreing Key* (FK)uma chave que vem de outra tabela para dentro da tabela Consulta, onde se guarda a data e horário da consulta além de suas chaves estrangeiras e primaria.

### **2.2.8. PHP**

De acordo com Converse e Park (2003), a linguagem PHP pode facilitar o desenvolvimento *Web,* o PHP é a sigla de *Hypertext Preprocessor* e é uma criação de *scripts* ao lado servidor que pode ser adicionado na linguagem HTML.

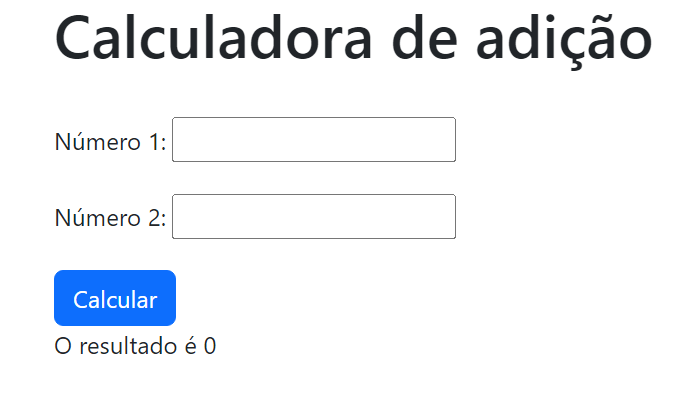
Para Juliano Niederauer (2017), todos os *scripts* escritos serão executados pelo servidor, que assim poderá ser exibido como uma página processada no navegador.

Segundo David Tanskley (2002), o PHP é utilizado para fazer um *back-end*, ou seja, uma comunicação ao banco de dados, a qual é um servidor que armazena informações. O PHP tem uma ótima interação ao banco de dados além de relacionar-se com outros bancos de dados.

Figura 10 - Código em PHP

Fonte: Do próprio autor, 2024.

Na figura acima, o código PHP demonstra uma calculadora de adição, em que os números digitados pela pessoa. O código irá verificar os números e logo após mostrar o resultado do cálculo desejado.

Figura 11 – Resultado do código em PHP

Fonte: Do próprio autor, 2024.

### **2.2.8. Mysql**

De acordo com Milani (2007), o MySQL é um servidor e um SGBD relacional. O MySQL foi desenvolvido para lidar com aplicações de pequeno e médio porte, hoje em dia, possui diversas características de um banco de dados de grande porte.

Para Bento (2013), o MySQL e o PHP possuem uma interação harmônica e são um dos principais conjuntos, tornando a *web* mais dinâmica. Com essa junção a uma facilidade no código além da interação com o banco de dados.

Figura 10 - Exemplo MySQL

Texto

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

* *create database*: Cria o seu banco de dados.
* *Create Table*: Cria uma tabela dentro do seu banco.
* *int*: Usados para guardar valores de números inteiros
* *vachar*: Usado para guardar tanto números quanto caracteres
* *not null*: Garante que essa coluna da tabela valores nulos.
* *Insert Into*: Pode-se adicionar um ou mais registros dentro da tabela relacionada.
* *Values*: Adiciona os valores de acordo com a ordem descrita no *Insert Into*.

Na figura 10, está sendo feita uma criação de um banco de dados que se chama “mundo” com uma tabela chamada “cliente”, em que será feita a inserção do cadastro de um novo dado no banco de dados. A figura 11 irá exibir o resultado da imagem anterior.

Figura 11 - Visualização MySQL

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

**2.2.9. Node.js**

De acordo com Moraes (2023), o Node.js é um utilizador em tempo real de JavaScript, o qual é constantemente utilizado em programações *web,* além da permissão de rodar aplicações fora do navegador *web*.

Como afirmado pelo Pereira (2018), ao adotar essa linguagem, beneficiará em muito se sua aplicação trabalha com processamento de arquivos e consiste em uma arquitetura não bloqueante, lhe fazendo ser simples e prática.

Conforme discutido por Duarte (2024), o Node.js não é uma linguagem, pois programamos utilizando o JavaScript, sendo mais caracterizada como uma plataforma de aplicação, permitindo o *full-stack* do JavaScript..

### **2.2.10. NPM**

Conforme o Pereira (2014), o NPM é um dos maiores gerenciadores de pacotes do mundo, tornando-se o gerenciador de pacotes do node.js a partir das suas primeiras versões, simplificando a vida dos desenvolvedores e fazendo vários projetos migrarem ao node.js.

### **2.2.11. Expo**

De acordo com Fuentes (2023), o Expo é uma ferramenta para desenvolver aplicativos tanto para Android quanto para iOS. O Expo permite realizar uma emulação no Android, iOS e também em um celular físico do aplicativo desenvolvido pelo desenvolvedor.

### **2.2.12. React**

Para Samy Silva (2021), o *React* é uma biblioteca de JavaScript que simplifica e acelera o desenvolvimento de interfaces de usuário em alto desempenho. As ferramentas da biblioteca possuem métodos práticos para o desenvolvimento web.

2.2.13. Bootstrap

Segundo Matos e Zabot (2020), o Bootstrap é uma ferramenta de código aberto que facilita o desenvolvimento de sites e aplicativos web. Ele tem modelos de design em HTML e CSS, e tendo componentes predefinidos responsivos.

2.2.14. React Native

Segundo Pinho e Escudelario (2020), o *React Native* é um *framework* que disponibiliza diversas ferramentas para agilizar o desenvolvimento de aplicações móveis, facilitando a criação de aplicativos de celulares.

Figura 12 - Exemplo código React Native

Texto

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Após essa imagem (Figura - 12) temos uma breve explicação sobre cada linha de código:

* **Image** e **StyleSheet** são importados da biblioteca **react-native**. Esses são componentes e módulos built-in do React Native.
* **HelloWave** é importado do módulo *@/components/HelloWave*. Este é um componente personalizado definido em outro lugar do projeto.
* **ParallaxScrollView** é importado do módulo *@/components/ParallaxScrollView*. Este é outro componente personalizado.
* **ThemedText** e **ThemedView** são importados dos módulos *@/components/ThemedText* e *@/components/ThemedView*, respectivamente. Esses também são componentes personalizados.

**Definição do componente HomeScreen**

O código define então uma exportação padrão, *HomeScreen*, que é um componente funcional. Este componente retorna um elemento JSX que representa a interface do usuário da tela inicial.

**Componente ParallaxScrollView**

O componente *ParallaxScrollView* é usado como o elemento de nível superior na árvore JSX. Este componente é provavelmente uma implementação personalizada de uma visão de rolagem que suporta efeitos de paralaxe.

**Propriedades do ParallaxScrollView**

O componente *ParallaxScrollView* recebe duas propriedades: *headerBackgroundColor* e *headerImage*. A propriedade *headerBackgroundColor* é um objeto que define as cores de fundo do cabeçalho para temas claros e escuros. A propriedade *headerImage* é um elemento *Image* que exibe a imagem do logotipo do React.

**Conteúdo do ParallaxScrollView**

Dentro do componente *ParallaxScrollView*, há quatro elementos *ThemedView* que contêm texto e outros elementos. Cada elemento *ThemedView* tem o seu estilo específico, sendo definido pela propriedade *style*.

**Elementos ThemedText**

Os elementos *ThemedText* são usados para exibir texto com estilos diferentes. Eles recebem uma propriedade *type* que define o estilo do texto, como título, subtítulo e entre outros.

**Elemento HelloWave**

O elemento *HelloWave* é um componente personalizado que é usado para exibir uma pequena animação de onda.

**Texto personalizado**

O texto dentro dos elementos *ThemedText* foi personalizado para exibir mensagens específicas para cada passo.

**Estilo**

O estilo dos elementos *ThemedView* e *ThemedText* é definido pelas propriedades *style* e *type*, respectivamente. Isso permite que os elementos sejam personalizados para se adequar ao design da aplicação.

E após tudo isso que foi passado, abaixo irá mostrar o projeto finalizado e já dentro de um aparelho celular.

Figura 13 - Exemplo finalizado React Native

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

A imagem acima (Figura 13), mostra a aplicação mobile rodando em um celular Android com o código já explicado anteriormente.

2.2.15. Wireframes

Em conformidade com Gonzzato e Costa (2011), os Wireframes são uma possibilidade de compreender as decisões que serão realizadas na etapa inicial, para testar quaisquer mudanças visuais que possam ocorrer no processo.

Figura 14 - Exemplo Wireframe baixa fidelidade

Diagrama, Forma

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Na imagem acima (Figura 14), vemos o Wireframe de baixa fidelidade onde é um pré-protótipo da criação final.

Figura 15 - Exemplo Wireframe de alta fidelidade

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Acima vemos o resultado esperado a esse exemplo projeto, o Wireframe de alta fidelidade, que seria uma tela de navegador, onde os desenvolvedores possam se basear, porém há possibilidade de sofrer alterações ao decorrer do projeto.

### **2.2.16. Unified Modeling Language**

Consoante a Guedes (2011), a UML (*Unified Modeling Language*) é a linguagem que foi adotada internacionalmente, e oferece diversos diagramas que especificam as características e os comportamentos de tal software desenvolvido.

Como afirma Booch (2006), UML é uma linguagem gráfica utilizada para a visualização e construção para a arquitetura de projetos de sistemas, além de determinar os seus itens em linguagens de programação.

Em concordância com Ramos (2006), a UML dependendo do projeto, traz um suporte para o indivíduo ou organização, por suas diversas metodologias, e apenas utilizando uma única linguagem de modelagem.

**2.2.18. Diagrama de caso de uso**

Para Guedes (2011), o diagrama de caso de uso é utilizado geralmente no período de levantamento e análise de requisitos em um sistema. O diagrama facilita na compreensão aos usuários para entender como o sistema irá funcionar.

Figura 16 - Exemplo diagrama Caso de uso

**Diagrama

Descrição gerada automaticamente**Fonte: Do próprio Autor, 2024.

Na imagem acima (Figura 16), podemos ver o ator “Cliente” que tem as seguintes ações que: pode fazer, visualizar a publicação e reagir a ela; já o “Administrador” pode apagar, modificar, pesquisar e inserir publicações com o manter que inclui todas essas ferramentas.

**2.2.17. Diagrama de Atividades**

Consoante Guedes (2011), o diagrama de atividade é utilizado para modelar atividades, sendo um método, algoritmo ou um processo inteiro, que sempre contêm ações, porém não precisam estar representadas.

Figura 17 - Exemplo diagrama de atividade

Diagrama

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Observamos que a figura acima (Figura 17) é um exemplo de diagrama de atividade em que o usuário desta ação precisa de um e-mail e senha já cadastrados para conseguir visualizar e cancelar suas compras.

**2.2.18. Diagrama de sequência**

Segundo Guedes (2011), esse diagrama determina a sequência que ocorre em um processo, apresentando quais métodos devem ser lançados em ordem para os atores e os objetos envolvidos.

Figura 18 - Exemplo diagrama de sequência

Diagrama

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

A figura 18 está apresentando a sequência dos métodos em ordem para a realização de um login em um sistema. O diagrama contém dois objetos (“Interface do sistema”, “Banco de Dados”) e um ator (“Cliente”).

**2.2.19. Diagrama de classes**

De acordo com Guedes (2011), este diagrama é muito utilizado e um dos mais importantes dentro da UML, servindo de apoio para os demais diagramas, onde define a estrutura que as classes do sistema irão utilizar.

Figura 19 - Exemplo diagrama de classes

Uma imagem contendo Tabela

Descrição gerada automaticamente

Fonte: Do próprio autor, 2024.

Como podemos ver na imagem acima (Figura 19), a classe em questão é chamada de “Criar\_Conta” onde contém 3 divisões, sendo a primeira para a descrição da classe, a segunda armazena os tipos de dados e a terceira é para listar os métodos da classe.

* #: Visibilidade protegida, permite que classes derivadas acessem a classe base onde facilita a herança entre elas.
* +: Visibilidade publica, significa que qualquer outra classe pode acessá-la, onde não há restrições de acesso.
* /: Significa que o atributo em questão sofre algum tipo de cálculo, e no caso da data, quando essa operação for realizada, o valor da data de abertura será utilizado pelo sistema.
* 0..1: No mínimo 0 e máximo 1, indica que os objetos da classe em questão, não precisa ter um relacionamento obrigatório.
* int: Indica que um valor numérico inteiro pode ser guardado.
* Date: É usado para representar e manipular datas e horas.
* string: Armazena valores textuais, onde se há uma sequência de caracteres.
* int:int: Significa que esses métodos recebem um número inteiro como parâmetro, que é uma variável local que pode ser ou já está declarada.

Após essa breve explicação, podemos ver que para esse ator possa abrir sua conta é preciso colocar um e-mail e senha, onde após confirmá-los é gerado seu ID único para diferenciá-lo, onde facilita a integração ao banco de dados, além de guardar a data e hora da criação de sua conta.

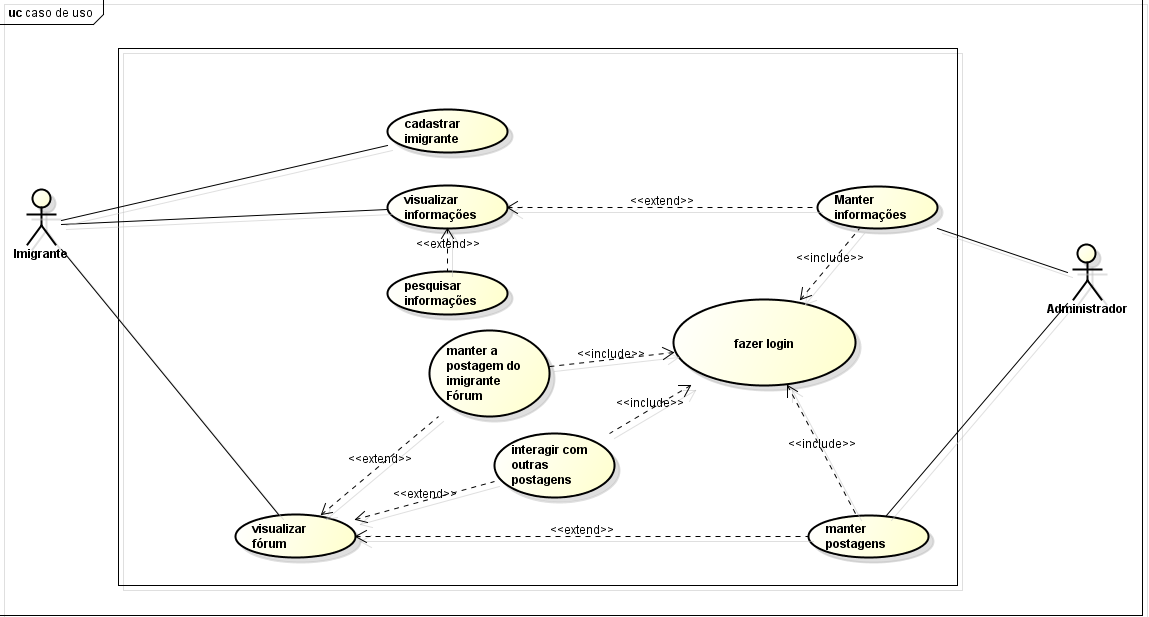
# 3. Desenvolvimento

Este segmento irá detalhar a metodologia de desenvolvimento do sistema Nuevos Caminos, fundamentando-se nos métodos de documentação UML. Além disso, incluirá imagens das interfaces do sistema e descrições de diagramas quando necessário.

# 3.1. Diagrama de Caso de Uso

Abaixo, há uma figura do Diagrama de Caso de Uso do sistema elaborado, na norma UML, revelando as relações dos atores com o sistema, que são: Imigrante e Administrador. Entretanto, as execuções que ao todo pode ser apresentada. Apresentam-se 9 elipses de casos de uso.

Figura 20 - Diagrama de Caso de Uso do Sistema

Fonte: Do próprio autor, 2024.

# 3.2. Documentação dos Casos de Usos

A seguir, apresenta-se a documentação dos casos de uso mostrados anteriormente, seguindo as orientações dos parâmetros da UML, destacando-se entre: Requisitos Funcionais, Requisitos Não Funcionais e Regras de negócio.

Os requisitos funcionais simbolizam as funcionalidades exigidas do sistema, sendo assim, é obrigatório para o seu funcionamento. Por outro lado, os requisitos não funcionais retratam as características necessárias pelo usuário, em geral, as aptidões relacionadas com a segurança, aparência, desempenho e a fiabilidade. Entretanto, as regras de negócio são peculiaridades fundamentais para cada cliente que deseja colocar em seu sistema.

Requisitos Funcionais do Imigrante:

* RF01: O imigrante poderá se cadastrar-se no sistema;
* RF02: O imigrante poderá visualizar informações;
* RF03: O imigrante poderá pesquisar informações;
* RF04: O imigrante poderá visualizar o fórum;
* RF05: O imigrante poderá manter a postagem do imigrante fórum;
* RF06: O imigrante poderá fazer login;
* RF07: O imigrante poderá interagir com outras postagens.

Requisitos Funcionais do Administrador:

* RF01: O administrador poderá manter as informações;
* RF02: O administrador poderá fazer login;
* RF03: O administrador poderá manter postagens;
* RF04: O administrador poderá visualizar fórum.

Requisitos Não Funcionais do sistema:

* RNF01: O site utilizará linguagens HTML, CSS, JavaScript e PHP;
* RNF02: A tecnologia que será utilizada no desenvolvimento do site irá ser a linguagem de banco de dados MySQL;
* RFN03: O sistema irá agir nos critérios UML;
* RFN04: O aplicativo utilizará a linguagem de interface React;
* RFN05: O aplicativo utilizará o framework Expo e React Native;
* RFN06: O aplicativo terá conectividade offline.

Regras de Negócio do Sistema:

* Somente imigrantes 18 anos ou mais podem utilizar;
* Os usuários devem fornecer informações pessoais precisas e atualizadas ao se registrar.;
* Políticas de privacidade claras devem ser divulgadas e aceitas pelos imigrantes;
* Apenas administradores podem gerenciar conteúdo sensível ou banir usuário;
* Todo conteúdo publicado pelos usuários deve ser revisado;
* Listagens de moradias devem ser verificados quanto à precisão e atualidade;
* Informações de contato dos locadores devem ser mantidas confidenciais e acessíveis apenas para usuários registrados;
* Informações sobre serviços essenciais devem ser atualizadas regularmente;
* Usuários devem seguir regras de etiqueta ao postar no fórum;
* Administradores têm o direito de remover postagens inadequadas;
* Usuários podem editar ou excluir seus próprios conteúdos;
* Comentários ofensivos ou falsos devem ser removidos;
* Usuários devem ter a oportunidade de fornecer feedback sobre suas experiências no site;
* Feedback deve ser analisado para melhorias contínuas no serviço.

Quadros retratados dos casos de usos:

**Documentação de casos de Uso de todos os atores**:

Quadro 1 – Descrição do caso de uso: Fazer Login

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Fazer Login |
| **Ator Principal** | Imigrante e administrador |
| **Resumo** | Neste caso de uso, é relatado o procedimento para a execução do login |
| **Pós-condição** | Para realizar o login, é necessário está cadastrado no sistema |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Solicitar o usuário e senha |
| 2. Preencher os dados para o login |  |
|  | 3. Analisar dados |
|  | 4. login finalizado |
| **Cenário Alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Avisar que os dados fornecidos estão incorretos |
|  | 2. Solicitação rejeitada |

Fonte: Do próprio autor, 2024.

**Documentação de casos de Uso do Imigrante**:

Quadro 2 – Descrição do casos de uso: Cadastrar Imigrante

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Cadastrar Imigrante |
| **Ator Principal** | Imigrante |
| **Resumo** | Neste caso de uso, relata o procedimento de cadastrar um Imigrante. |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Solicitar dados do imigrante |
| 2. Preencher os dados para a criação da conta. |  |
|  | 3. Analisar dados |
|  | 4. Cadastro finalizado |
| **Restrições/Validação** | 1. É necessário o imigrante ter 18 anos ou mais |
| **Cenário Alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Avisar que os dados fornecidos estão incorretos |
|  | 2. Solicitação rejeitada |

Fonte: Do próprio autor, 2024.

Quadro 3 – Descrição do casos de uso: Visualizar Informação

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Visualizar Informação |
| **Ator Principal** | Imigrante |
| **Resumo** | Neste caso de uso, é relatado o procedimento para visualizar informações |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Visualizar informações |  |
| **Cenário alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar informações |  |
|  | 2. Analisar pesquisa |
|  | 3. Exibir resultado |

Fonte: Do próprio autor, 2024.

Quadro 4 - Documentação do caso de uso: Pesquisar Informações

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Pesquisar Informações |
| **Ator Principal** | Imigrante |
| **Resumo** | Neste caso de uso, é relatado o procedimento para pesquisar informações |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Exibir informações |
| 2. Pesquisar informações |  |
|  | 3. Analisar pesquisa |
|  | 4. Exibir resultado |
| **Cenário alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Pesquisar informações |  |
|  | 2. Analisar pesquisa |
|  | 3. Avisar que não se encontra no site |

Fonte: Do próprio Autor, 2024.

Quadro 5 - Descrição do caso de uso: Visualizar Fórum

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Visualizar Fórum |
| **Ator Principal** | Imigrante |
| **Resumo** | Neste caso de uso, é relatado o procedimento para visualizar fórum |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Visualizar fórum |  |
| **Cenário alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Manter postagem do imigrante |  |
|  | 2. Avisar que é necessário fazer login |
| **Cenário alternativo II** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Interagir com outros postagens |  |
|  | 2. Avisar que é necessário fazer login |

Fonte: Do próprio autor, 2024.

Quadro 6 - Descrição do caso de uso: Manter a postagem do imigrante Fórum

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Manter a postagem do imigrante Fórum |
| **Ator Principal** | Imigrante |
| **Resumo** | Neste caso de uso, é relatado o procedimento para manter a postagem do imigrante fórum |
| **Pós-condição** | Para Manter a postagem do imigrante fórum, |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Visualizar fórum |  |
| **Cenário alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Manter postagem do imigrante |  |
|  | 2. Avisar que é necessário Fazer login |
| **Cenário alternativo II** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Interagir com outros postagens |  |
|  | 2. Avisar que é necessário Fazer login |

Fonte: Do próprio autor, 2024.

Quadro 7 – Documentação do caso de uso: Interagir com outras postagens

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Interagir com outras postagens |
| **Ator Principal** | Imigrante |
| **Resumo** | Neste caso de uso, é relatado o procedimento para interagir outras postagem |
| **Pós-condição** | Para interagir as publicações, é necessário Fazer Login |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. exibir postagens |
| 2. Interagir postagem |  |

Fonte: Do próprio autor, 2024.

**Documentação de casos de Uso do Administrador**:

Quadro 8 - Documentação do caso do uso: Manter Informação

|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Manter informação |
| **Ator Principal** | Administrador |
| **Resumo** | Neste caso de uso, é relatado o procedimento para manter informação |
| **Pós-condição** | Para Manter informação, é necessário fazer login, |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Exibir Informação |
| 2. Manter Informação |  |

Fonte: Do próprio autor, 2024.

Quadro 9 - Documentação de caso de uso: Manter postagem

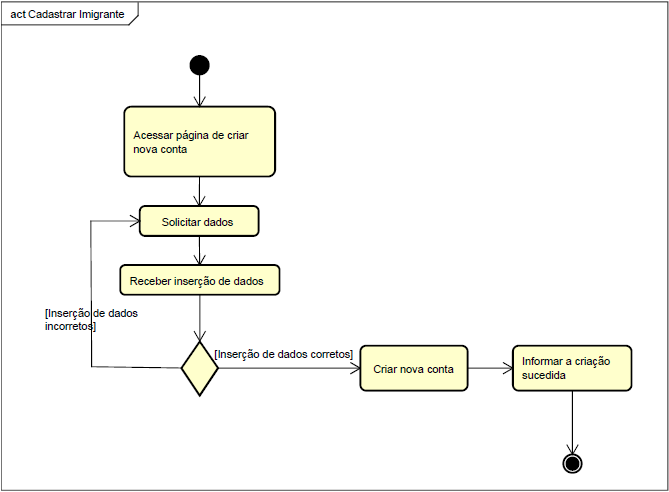
|  |  |
| --- | --- |
| **Nome do Caso de Uso** | Manter postagem |
| **Ator Principal** | Administrador |
| **Resumo** | Neste caso de uso, é relatado o procedimento para manter postagem |
| **Pós-condição** | Para Manter postagem, é necessário fazer login, |
| **Cenário Principal** | |
| **Ações do Ator** | **Ações do Sistema** |
|  | 1. Exibir postagens |
| 2. Manter postagem |  |
| **Cenário alternativo I** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Manter postagem do imigrante |  |
|  | 2. Avisar que é necessário Fazer login |
| **Cenário alternativo II** | |
| **Ações do Ator** | **Ações do Sistema** |
| 1. Interagir com outros postagens |  |
|  | 2. Avisar que é necessário Fazer login |

Fonte: Do próprio autor, 2024.

# 3.3. Diagrama de Atividade

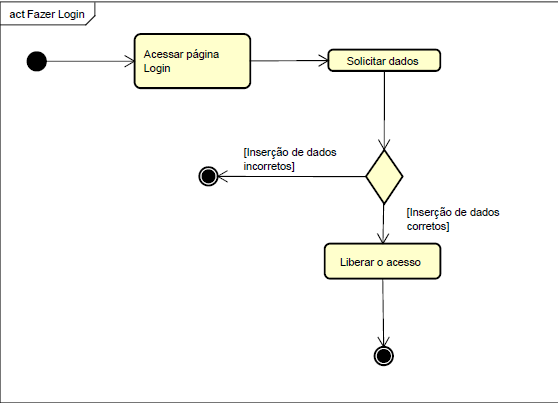
O diagrama de atividade pode representar as etapas do processo completo do projeto, fornecendo detalhes das ações presentes no sistema. Cada elemento do diagrama de atividade demostra as execuções de cada afazer de funcionalidade do sistema.

Figura 21 - Diagramade Atividade: Cadastrar Imigrante



Fonte: Do próprio autor, 2024.

Figura 22 - Diagrama de Atividade: Fazer login

****

Fonte: Do próprio autor, 2024.

Figura 23- Diagrama de Atividade: Interagir com outras postagens

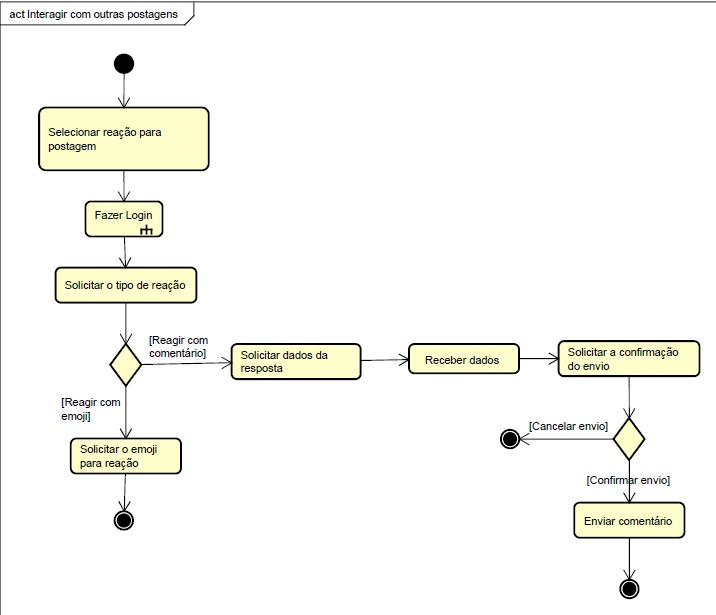
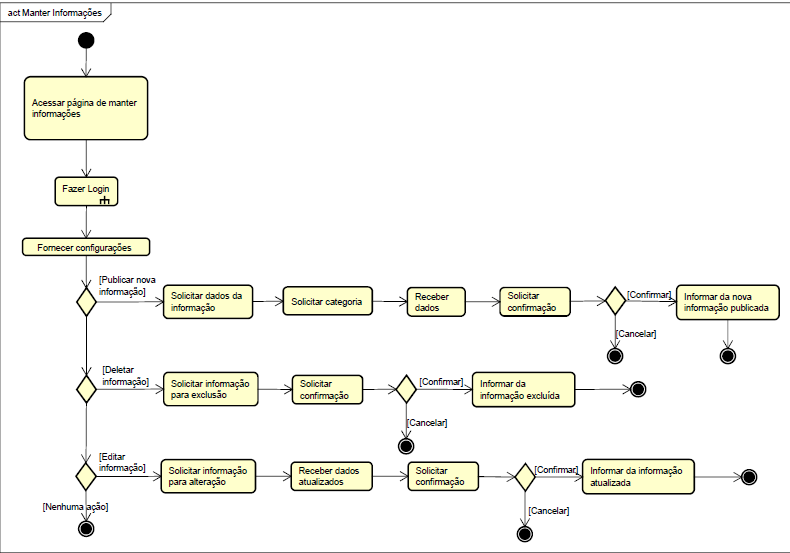
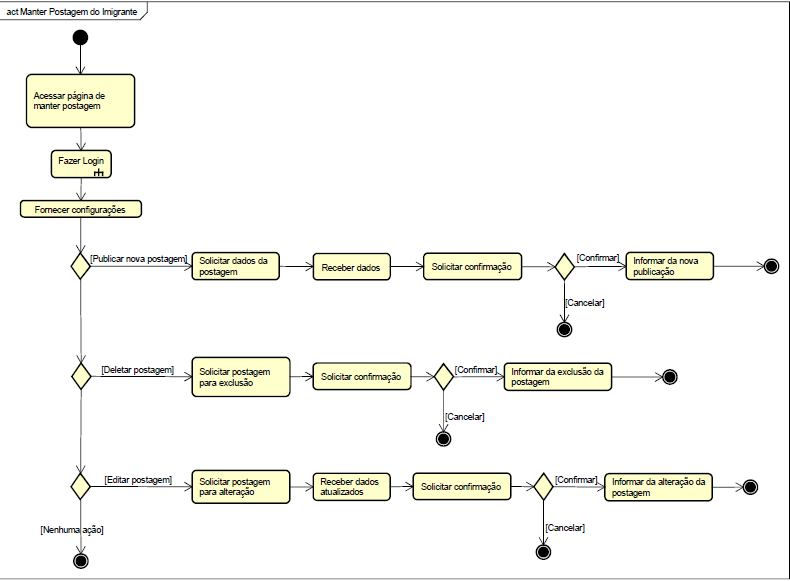
****Fonte: Do próprio autor, 2024.

Figura 24 - Diagrama de Atividade: Manter informações

****

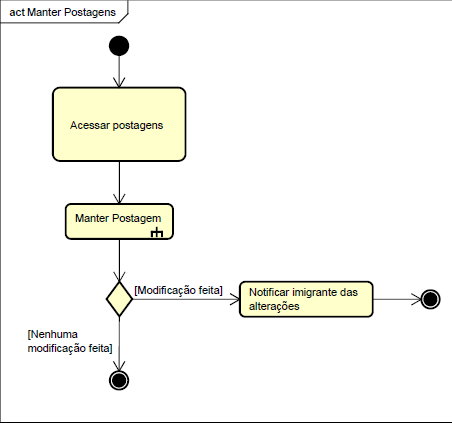
Fonte: Do próprio autor, 2024.

Figura 25 - Diagrama de Atividade: Manter Postagem do Imigrante

****

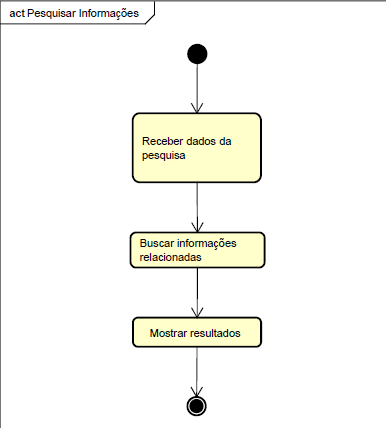
Fonte: Do próprio autor, 2024.

Figura 26 - Diagrama de Atividade: Manter Postagens

****

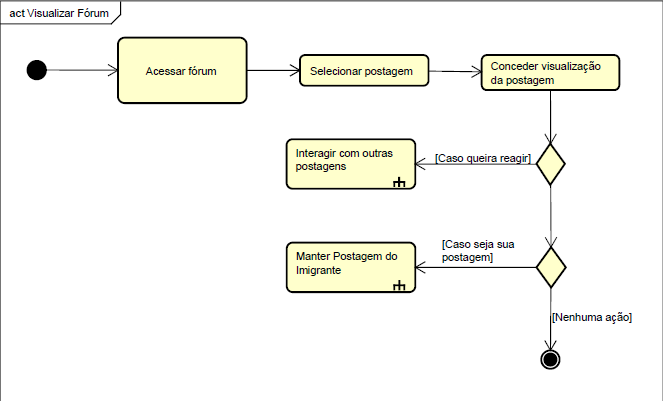
Fonte: Do próprio autor, 2024.

Figura 27 - Diagrama de Atividade: Pesquisar Informações

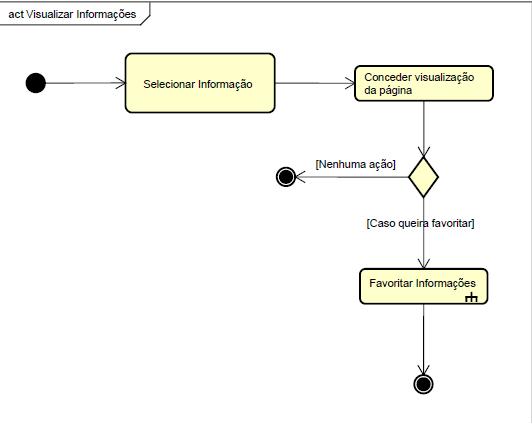
****

Fonte: Do próprio autor, 2024.

Figura 28 - Diagrama de Atividade: Visualizar Fórum



Fonte: Do próprio autor, 2024.

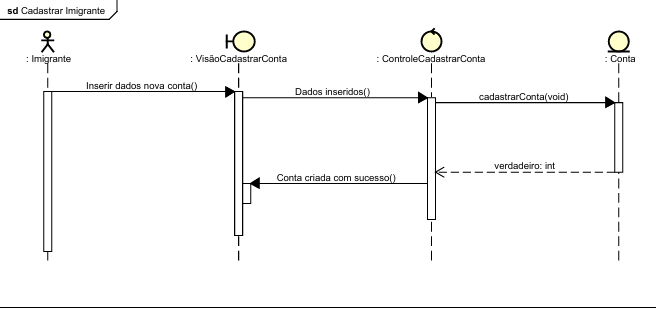
Figura 29 - Diagrama de Atividade: Visualizar Informações 

Fonte: Do próprio autor, 2024.

# 3.4. Diagrama de Sequência

A seguir, os diagramas de sequência que serão apresentados permitem observar o funcionamento de somente um único cenário, prosseguindo em formato de linha temporal e as ações relacionadas á realização de uma função específica. No projeto, há um total de 9 diagramas de sequência que retratam as execuções dos atores e da interface, as interfaces e classes mostram a interação com o banco de dados. Sendo assim, demonstra detalhadamente a operação de todos os atores que estão presentes no sistema e organiza em etapas cada interação para o funcionamento de uma ação exigida do sistema.

Figura 30 – Diagrama de Sequência: Cadastrar Imigrante



Fonte: Do próprio autor, 2024.

Figura 31 - Diagrama de Sequência: Fazer Login

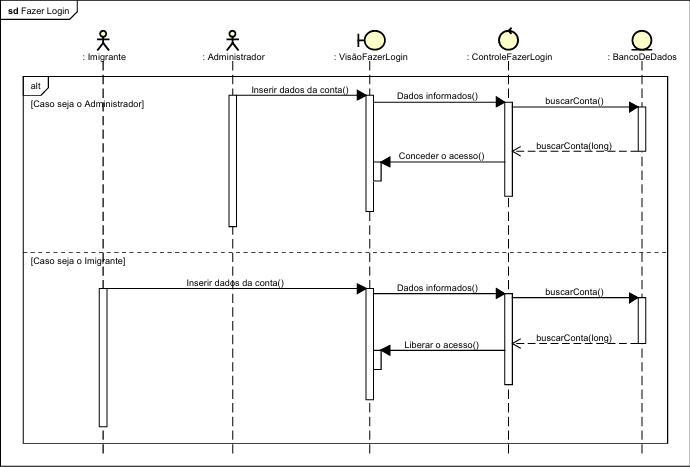
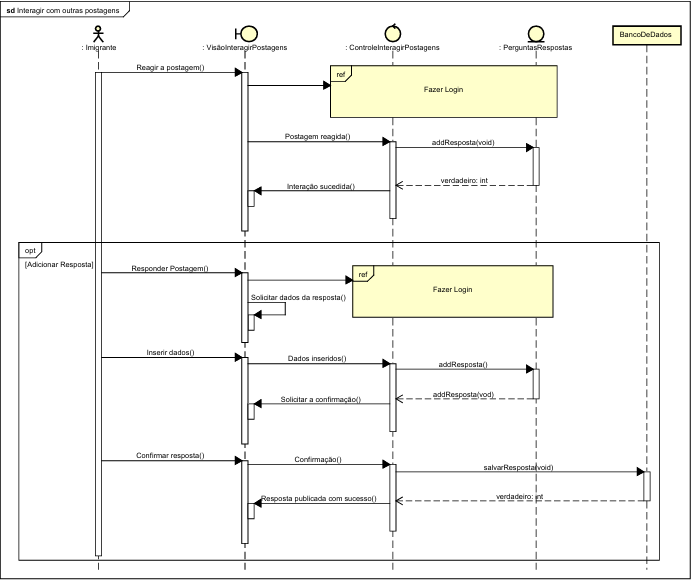
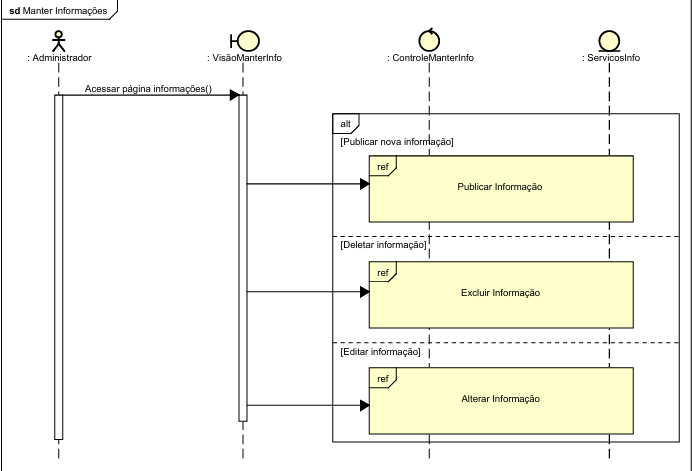
****Fonte: Do próprio autor, 2024.

Figura 32 - Diagrama de Sequência: Interagir com outras postagens

****

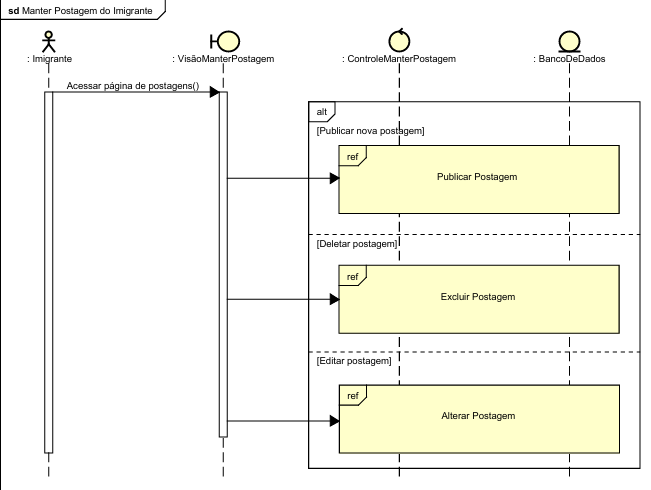
Fonte: Do próprio autor, 2024.

Figura 33 - Diagrama de Sequência: Manter Informações

****

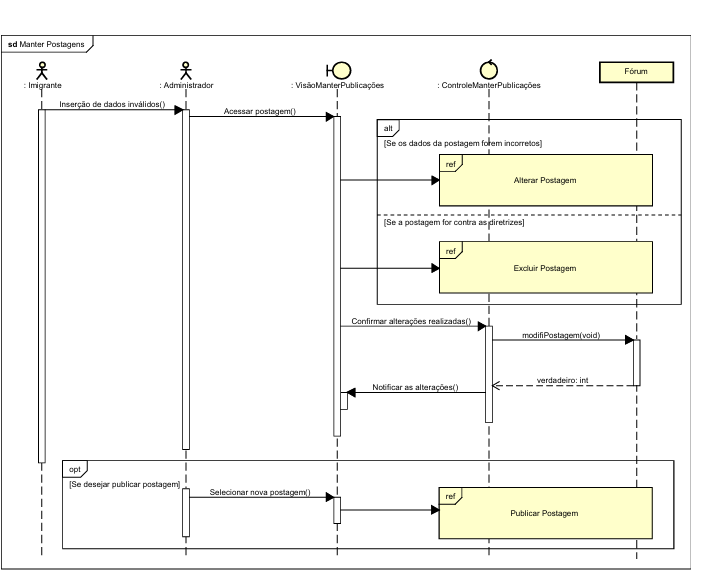
Fonte: Do próprio autor, 2024.

Figura 34 - Diagrama de Sequência: Manter Postagem do Imigrante

****

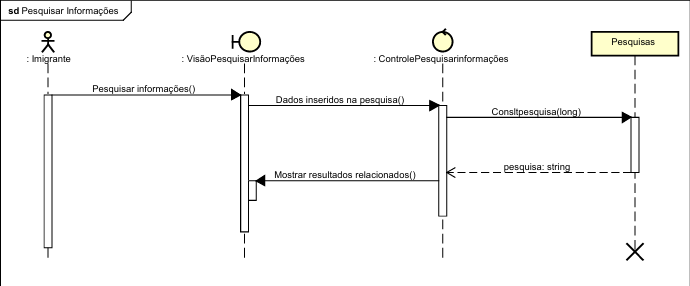
Fonte: Do Próprio autor, 2024.

Figura 35 - Diagrama de Sequência: Manter Postagens

****

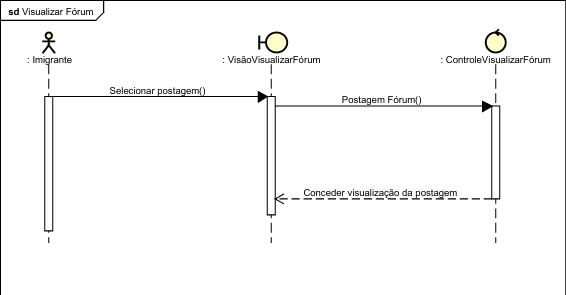
Fonte: Do próprio autor, 2024.

Figura 36 - Diagrama de Sequência: Pesquisar Informação

****

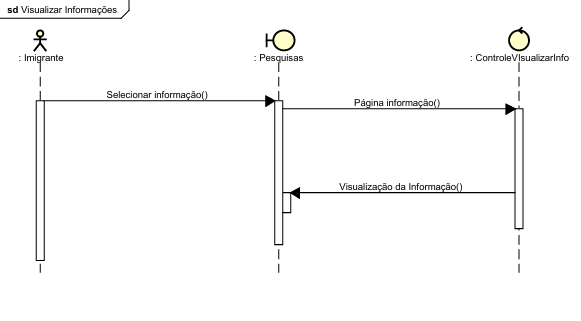
Fonte: Do próprio autor, 2024.

Figura 37 - Diagrama de Sequência: Visualizar Fórum

****

Fonte: Do próprio autor, 2024.

Figura 38 - Diagrama de Sequência: Visualizar Informações

****

Fonte: Do próprio autor, 2024.

**REFERÊNCIAS**

ALBINO, Igor Cacerez**. Dominando O Css**. [S. I.]: [s.n.] Clube de Autores, 2023.

BAZZI, Cláudio Leones. **Introdução à Banco de dados**. 22. ed. Curitiba: Ed. UTFPR. 2013.

BENTO, Junior Evaldo. **Desenvolvimento Web com PHP e MySQL.** 1. ed. São Paulo: Casa do Código, 2013.

BOOCH, Grady. **UML:** Guia do usuário. 2. ed. Rio de Janeiro: Campus Editora. 2006. E-book.

CARRIL, Marly. **HTML – Passo a Passo.** Clube de Autores, 2012.

DUARTE, Luiz. **Node.js e Microservices:** Um guia prático. 5. ed. LuizTools. 2024. E-book

DUCKETT, Jon. **HTML e CSS:** projete e construa websites. 1. ed. Rio de Janeiro: Alta Books, 2016.

ELMASRI, Ramez. **Sistema de banco de dados**.São Paulo: Pearson Addison Wesley, 2005.

ESCULADEIRO, Bruna; PINHO, Diego. **React Native: Desenvolvimento de aplicativos mobile com React.** 1. Ed. São Paulo: Editora Casa do Código, 2020.

FLANAGAN, David. **Javascript:** O guia definitivo. 6. ed. Rio de Janeiro: BOOKMAN Editora. 2012.

FLATSCHART, Fábio. **HTML 5 – Embarques Imediato**. Brasport, 2011.

GUEDES, Gilleanes Thorwald Araujo. **UML 2:** Uma abordagem prática. 3. ed. São Paulo: Novatec Editora. 2011. E-book.

MATOS, Ecivaldo de Souza; ZABOT, Diego. **Aplicativos com Bootstrap e Angular: Como Desenvolver Apps Responsivos: como desenvolver apps reponsivos**.1. ed. São Paulo: Editora Érica, 2020.

MILANI, André. **MySQL – Guia do Programador.** 1. ed. São Paulo: Novatec Editora, 2007.

MORAES, Willian Bruno. **Construindo aplicações com NodeJS**. 1. ed. São Paulo: Novatec Editora Ltda. 2015.

MORRISON, Michel. **Use a cabeça!** JavaScript. 1. ed. Rio de janeiro: Alta Books. 2020.

PEREIRA, Caio Ribeiro. **Node.js:** Aplicações web real-time com Node.js. 1. ed. Casa do Código. 2018. E-book.

PRESCOTT, Preston. **Programação em Javascript**. Babelcube Inc. 2016.

PRESCOTT, Preston. **SQL para Iniciantes**. [S.E], [S.E]: babelcube inc., 2015.

QUIERELLI, David Antonio. **Criando sites com html-css-php**. Clube de Autores, 2012.

RAMOS, Ricardo Argento. **Treinamento prático em UML**. 1. ed. São Paulo: Digerati Books. 2006. E-book.

SAMY, Maurício Silva. **React Aprenda Praticando: Desenvolva aplicações web reais com uso da biblioteca React e de seus módulos auxiliares.** 1. Ed. São Paulo: Editora Novatec, 2021.

ZEMEL, Tárcio. **CSS Eficiente:** técnicas e ferramentas que fazem a diferença nos seus estilos. São Paulo: Casa do Código, 2015.

COSTA, Karla da Cruz; GONZZATO, Rodrigo Fresse. **Arquitetura de informação sem wireframe**. **Perspectivas em Gestão e Conhecimento**, v. 1, p. 160-181, 2011.

AGÊNCIA BRASIL, **Números de novos imigrantes cresce em 24,4% no Brasil em 10 anos.** Agência Brasil, 2021.Disponível em: <https://agenciabrasil.ebc.com.br/geral/noticia/2021-12/numero-de-novos-imigrantes-cresce-244-no-brasil-em-dez-anos>. Acesso em: 27 mar. 2024.

BEPAYBANK, **Imigração no Brasil: os desafios financeiros e a dificuldade no envio de remessas.** G1globo, 2024.Disponível em: <https://g1.globo.com/sp/campinas-regiao/especial-publicitario/bepay-bank/bepay-bank-transferencias-internacionais/noticia/2024/03/11/imigracao-no-brasil-os-desafios-financeiros-e-a-dificuldade-no-envio-de-remessas.ghtml>**.** Acesso em: 29 mar. 2024.

CLOUD. **O que é um banco de dados relacional?** [S. I.]:Cloud, [S. D.]. Disponível em: <https://cloud.google.com/learn/what-is-a-relational-database?hl=pt-br>. Acesso em: 07 mai. 2024.

EXPO. **Introdução**. Expo.dev. Disponível em: [Introduction - Expo Documentation](https://docs.expo.dev/get-started/introduction/). Acesso em: 29 ago. 2024.

IBM. **O que é um banco de dados relacional?**. [S. I.]: IBM, [S. D.]. Disponível em: <https://www.ibm.com/br-pt/topics/relational-databases>. Acesso em: 06 mai. 2024.

MICROSOFT. **O que é um Banco de Dados SQL?**. Microsoft. Disponível em: <https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-sql-database>. Acesso em: 15 mai. 2024.

NPM. **Sobre o NPM**. Npm. Disponível em: [About npm | npm Docs (npmjs.com)](https://docs.npmjs.com/about-npm). Acesso em: 28 mar. 2024.

PESSONI, Mara. **Os maiores desafios enfrentados pelos imigrantes**. Migalhas, 2023. Disponível em: <https://www.migalhas.com.br/depeso/394964/os-maiores-desafios-enfrentados-pelos-imigrantes>. Acesso em: 28 mar. 2024.

ORACLE. **O que é um banco de dados relacional (RDBMS)?**. [S. I.]: Oracle, [S. D.]. Disponível em: <https://www.oracle.com/br/database/what-is-a-relational-database/>. Acesso em: 06 mai. 2024.

Converse, Tim; Park, Joyce. PHP - a Bíblia. 2. ed. São Paulo: Elsevier Editora Ltda., 2003.

Niederauer, Juliano. Desenvolvendo websites com PHP: Aprenda a criar websites dinâmicos e interativos com PHP e banco de dados. 3. ed. São Paulo: Novatec Editora, 2017.

Tanskley, David. Como criar web pages rápidas e eficientes usando php e Mysql. 1. ed. Rio de Janeiro: Editora Ciência Moderna, 2002.