SENAI

GABRIEL GALVÃO DE OLIVEIRA MARQUES

**BRASIL EM VIAGEM**

**UM BLOG PARA COMPARTILHAR**

**SUAS EXPERIÊNCIAS**

Curitiba – PR

2023

GABRIEL GALVÃO DE OLIVEIRA MARQUES

**BRASIL EM VIAGEM**

**UM BLOG PARA COMPARTILHAR**

**SUAS EXPERIÊNCIAS**

Trabalho avaliativo apresentado ao curso Técnico em desenvolvimento de sistemas. Sistema Fiep, unidade Dr. Celso Charuri, como requisito parcial para obter o diploma Técnico em Desenvolvimento de sistemas.

Orientador: Prof. Gustavo Vinicius.

Curitiba – PR

2023

Sumário

[1 INTRODUÇÃO: 5](#_Toc137997006)

[JUSTIFICATIVA: 5](#_Toc137997007)

[1.2 OBJETIVOS: 5](#_Toc137997008)

[1.21 OBJETIVO GERAL: 5](#_Toc137997009)

[1.22 OBJETIVO ESPECÍFICO: 6](#_Toc137997010)

[2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA 6](#_Toc137997011)

[2.1 REQUISITOS FUNCIONAIS 6](#_Toc137997012)

[2.2 ANÁLISE DE REQUISITOS 7](#_Toc137997013)

[2.3 REQUISITOS NÃO-FUNCIONAIS 9](#_Toc137997014)

[3 DESENVOLVIMENTO DO SISTEMA E MÉTODOS 11](#_Toc137997015)

[3.1 PROCESSO DE DESENVOLVIMENTO: 11](#_Toc137997016)

[3.2 TECNOLOGIAS UTILIZADAS: 12](#_Toc137997017)

[3.3 MÉTODOS 12](#_Toc137997018)

[4 TELAS DO SISTEMA 16](#_Toc137997019)

[5 DIAGRAMAS 18](#_Toc137997020)

[6 CONCLUSÃO 18](#_Toc137997021)

# 1 INTRODUÇÃO:

O projeto "Brasil em viagem" é um **blog de** **conteúdos voltados apenas para turismo no Brasil**, com o objetivo principal de proporcionar a troca de experiências, receitas e conhecimentos sobre os destinos no Brasil.

Sendo um **protótipo** de uma plataforma de uso colaborativo, seus usuários poderão criar tipos de publicações diferentes (artigos, roteiros de viagem e receitas). Para utilização desse protótipo é necessária a criação de uma conta por parte do usuário.

O Brasil em Viagem possui uma navegação fácil, se destacando pela usabilidade que ele proporciona ao usuário, sendo intuitivo com páginas simples e de uso dinâmico.

Embora seja atualmente um protótipo em desenvolvimento, o objetivo final é criar uma plataforma totalmente funcional e interativa, onde os usuários possam interagir entre si, comentar nas postagens, compartilhar nas redes sociais e até mesmo contribuir com sugestões e edições nos artigos existentes.

O projeto "Brasil em Viagem" visa despertar o espírito aventureiro dos leitores, fornecendo um espaço digital onde eles possam explorar, aprender e se inspirar para suas próprias viagens. Com a participação e colaboração dos usuários, o blog poderá se tornar uma referência confiável no mundo das viagens.

Palavras chave: Protótipo, blog de conteúdos voltados para turismo apenas no Brasil.

# JUSTIFICATIVA:

Como reforçado anteriormente, hoje no Brasil não existem ferramentas que proporcionem uma experiência abrangente para com as necessidades e dinamismo que o usuário necessita.

# 1.2 OBJETIVOS:

# 1.21 OBJETIVO GERAL:

Trazer informações pertinentes sobre todo o Brasil de forma intuitiva e assertiva para o leitor.

Curitiba – PR

2023

# 1.22 OBJETIVO ESPECÍFICO:

* Pesquisar e analisar plataformas existentes de blogs de viagens para identificar recursos relevantes e melhores práticas.
* Definir os requisitos e funcionalidades principais do protótipo do blog de viagens colaborativo.
* Projetar e desenvolver a interface do usuário do protótipo, garantindo uma experiência intuitiva e agradável para os usuários.
* Implementar recursos de criação de publicações, como artigos, roteiros de viagem e receitas culinárias, com opções de edição e exclusão.
* Permitir que os usuários criem perfis pessoais, interajam entre si, comentem nas postagens e compartilhem nas redes sociais.
* Realizar testes de usabilidade para identificar e corrigir possíveis problemas de interface e funcionalidade.

# 2 REQUISITOS FUNCIONAIS E NÃO FUNCIONAIS DO SISTEMA

# 2.1 REQUISITOS FUNCIONAIS

|  |  |
| --- | --- |
| Número | Requisito funcional |
| RF1 | Operação de cadastro |
| RF2 | Operação de login |
| RF3 | Publicação de artigos |
| RF4 | Publicação de roteiros de viagem |
| RF5 | Publicação de receitas |
| RF6 | Exclusão de artigos |
| RF7 | Exclusão de roteiros de viagem |
| RF8  RF9  R10  R11  R12 | Exclusão de receitas  Alterar dados do perfil  Busca de artigos  Busca de roteiros de viagem  Busca de receitas |
|  |  |

# 2.2 ANÁLISE DE REQUISITOS

RF1: Operação de cadastro

* Descrição: Permite que os usuários se cadastrem no sistema, fornecendo informações básicas para criar uma conta.
* Observações: É importante definir quais campos são necessários no cadastro, como nome, e-mail, senha, etc. Além disso, deve-se considerar a validação dos dados fornecidos e a garantia da segurança das informações.

RF2: Operação de login

* Descrição: Permite que os usuários façam login no sistema utilizando suas credenciais.
* Observações: É necessário implementar a autenticação de usuário, verificando se as credenciais estão corretas e concedendo acesso ao sistema. Também é importante considerar a segurança, como a proteção contra ataques de força bruta ou uso de medidas como autenticação em dois fatores.

RF3: Publicação de artigos

* Descrição: Permite que os usuários criem e publiquem artigos no sistema.
* Observações: Deve-se definir quais informações serão solicitadas para a criação do artigo, como título, conteúdo, categoria, tags, entre outros. Além disso, pode ser interessante implementar recursos adicionais, como a possibilidade de adicionar imagens, vídeos ou formatar o texto.

RF4: Publicação de roteiros de viagem

* Descrição: Permite que os usuários criem e publiquem roteiros de viagem no sistema.
* Observações: É necessário definir quais informações serão necessárias para criar um roteiro de viagem, como destino, duração, descrição dos pontos turísticos, entre outros. Também pode ser interessante permitir a inclusão de imagens, mapas ou itinerários.

RF5: Publicação de receitas

* Descrição: Permite que os usuários criem e publiquem receitas no sistema.
* Observações: Deve-se definir quais informações serão solicitadas para a criação da receita, como nome, ingredientes, modo de preparo, tempo de cozimento, entre outros. Também pode ser útil permitir a adição de fotos do prato pronto.

RF6: Exclusão de artigos

* Descrição: Permite que os usuários excluam artigos previamente publicados por eles.
* Observações: Deve-se implementar um mecanismo que permita aos usuários excluir seus próprios artigos de forma segura. Pode ser necessário definir regras adicionais, como permitir a exclusão apenas se não houver comentários associados ao artigo.

RF7: Exclusão de roteiros de viagem

* Descrição: Permite que os usuários excluam roteiros de viagem previamente publicados por eles.
* Observações: Semelhante à exclusão de artigos, é necessário implementar um mecanismo seguro para que os usuários possam excluir seus próprios roteiros de viagem.

RF8: Alterar dados do perfil:

* Descrição: Permite que os usuários façam alterações em seus dados de perfil, como nome, e-mail, senha, foto de perfil, entre outros.
* Observações: Deve-se fornecer uma interface intuitiva para que os usuários possam modificar suas informações pessoais. É importante garantir a segurança ao realizar alterações, como a validação das senhas e a proteção contra ações não autorizadas.

RF9: Exclusão de receitas

* Descrição: Permite que os usuários excluam receitas previamente publicadas por eles.
* Observações: Deve-se implementar um mecanismo seguro para que os usuários possam excluir suas próprias receitas. Assim como nas exclusões anteriores, podem ser necessárias regras adicionais, como permitir a exclusão apenas se não houver comentários associados à receita

RF10: Busca de artigos:

* Descrição: Permite que os usuários realizem buscas por artigos com base em critérios como título, conteúdo, categoria, tags, etc.
* Observações: É necessário implementar um mecanismo de busca eficiente que permita aos usuários localizar artigos relevantes. Considerar a otimização da pesquisa e a apresentação de resultados relevantes.

R11: Busca de roteiros de viagem:

* Descrição: Permite que os usuários realizem buscas por roteiros de viagem com base em critérios como destino, duração, pontos turísticos, etc.
* Observações: Assim como na busca de artigos, é necessário implementar um mecanismo de busca eficiente para roteiros de viagem. Considerar critérios de pesquisa relevantes para esse tipo de conteúdo.
* R12: Busca de receitas:
* Descrição: Permite que os usuários realizem buscas por receitas com base em critérios como nome, ingredientes, tempo de cozimento, etc.
* Observações: Semelhante às buscas anteriores, é necessário implementar um mecanismo de busca eficiente para receitas, considerando critérios relevantes para esse tipo de conteúdo culinário.

# 2.3 REQUISITOS NÃO-FUNCIONAIS

|  |
| --- |
| REQUISITOS NÃO FUNCIONAIS |
| Operação de cadastro: |
| * RNF1: Segurança: O sistema deve garantir a proteção dos dados pessoais durante o processo de cadastro. |
| * RNF2: Usabilidade: A interface de cadastro deve ser intuitiva e de fácil compreensão para os usuários. |
| * RNF3: Desempenho: O tempo de resposta do sistema durante o cadastro não deve exceder um limite aceitável. |
| Operação de login: |
| * RNF4: Segurança: O sistema deve implementar mecanismos de autenticação seguros para proteger as informações de login dos usuários. |
| * RNF5: Disponibilidade: O sistema de login deve estar disponível e acessível para os usuários sempre que necessário. |
| * RNF6: Desempenho: O tempo de resposta do sistema durante o processo de login deve ser rápido e eficiente.   Publicação de artigos, roteiros de viagem e receitas:   * RNF7: Desempenho: O sistema deve ser capaz de lidar com um grande número de publicações simultaneamente, sem comprometer o desempenho. * RNF8: Escalabilidade: O sistema deve ser capaz de lidar com o crescimento do número de publicações ao longo do tempo, sem diminuir sua eficiência. * RNF9: Armazenamento: O sistema deve ter capacidade de armazenamento adequada para suportar todas as publicações de artigos, roteiros de viagem e receitas. * Exclusão de artigos, roteiros de viagem e receitas: * RNF10: Integridade dos dados: A exclusão de publicações deve ser realizada de forma segura, garantindo que os dados sejam removidos corretamente do sistema. * RNF11: Auditoria: O sistema deve manter um registro de exclusões, permitindo rastrear quem realizou a exclusão e quando.   Alterar dados do perfil:   * RNF12: Segurança: As alterações nos dados do perfil devem ser feitas apenas pelo usuário autenticado, garantindo a privacidade e a integridade das informações pessoais. * RNF13: Consistência: As alterações realizadas no perfil devem ser refletidas corretamente em todo o sistema, evitando informações desatualizadas ou inconsistentes.   Inserir informações adicionais e alterar informações adicionais:   * RNF14: Usabilidade: O sistema deve fornecer interfaces intuitivas e de fácil utilização para inserção e alteração de informações adicionais. * RNF15: Validação: O sistema deve validar as informações adicionais inseridas, garantindo que sejam consistentes e corretas.   Busca de artigos, roteiros de viagem e receitas:   * RNF16: Desempenho: O sistema deve ser capaz de realizar buscas eficientes e rápidas nas publicações, independentemente do número de registros existentes. * RNF17: Precisão: O sistema deve retornar resultados relevantes e precisos com base nos critérios de busca especificados pelo usuário. |

# 3 DESENVOLVIMENTO DO SISTEMA E MÉTODOS

# 3.1 PROCESSO DE DESENVOLVIMENTO:

Março: back-end 10%, front-end 20%.

Abril: back-end 30%, front-end 40%.

Maio: back-end 30%, front-end 80%.

Junho: back-end 65%, front-end 100%.

Curitiba – PR

2023

Março, após a separação da primeira equipe, qual eu e mais uma integrante estávamos apalavrados, eu trouxe a ideia ao orientador de produzir ou um “concorrente do Pinterest” ou uma “rede social voltada ao turismo” para meu trabalho, e após a aprovação do mesmo apresentei a ideia a outra integrante do primeiro grupo, qual se uniu a mim para o desenvolvimento. Eu e ela havíamos entrado em consentimento de que um faria o back-end (eu), e ela faria o front-end e ambos fariam a documentação.

Já em abril, após as aulas de front-end fornecidas pelo orientador, a outra integrante ainda apresentava dificuldades na produção do HTML e CSS. Então me propus paralisar a entrega do back-end para repartir as tarefas dela no front-end. Fizemos a modelagem das telas que precisaríamos, e para auxiliá-la, criei dois arquivos com a padronização de design que o site teria, “base.html” e “base.css”.

Já em Maio, cobro maior desempenho por parte da outra integrante da equipe afinal ela havia me entregado 4 telas copiadas de materiais disponíveis na internet, e não havia encaixado nos arquivos de base que fiz para auxiliá-la, sendo um trabalho extra que acabava tomando grande parte do meu tempo. Após a situação ficar insustentável, decido me separar da mesma comunicando o professor orientador e claro ela. O professor orientador responsável por coordenar nossos projetos sugeriu para que eu alterasse a ideia de um projeto que estava atrasado faltando apenas 22 dias para a entrega final, e realocou a integrante em outro grupo.

No mês da entrega, passei a me dedicar integralmente para a produção do projeto, porém, devido a todas as adversidades citadas anteriormente ele será apenas um protótipo de um blog voltado a turismo.

3.2 TECNOLOGIAS UTILIZADAS:

Foram utilizadas as seguintes tecnologias:  
Front-end: HTML, CSS e JavaScript;

Back-end: PHP; Xampp (Servidor Apache).

# 3.3 MÉTODOS

Método: **getConnection()**.

Descrição: Este método estabelece uma conexão com o banco de dados MySQL e retorna um objeto de conexão PDO (PHP Data Object).

Modificador de Acesso: **public**;

Retorno: Objeto de conexão PDO ou **null** em caso de erro na conexão.

Curitiba – PR

2023

Etapas do Método:

1. Criação das variáveis privadas **$username**, **$password**, **$db\_name** e **$host**, que armazenam as informações necessárias para a conexão com o banco de dados.
2. Inicialização da variável **$conn** com **null**.
3. No bloco **try-catch**, uma tentativa de conexão com o banco de dados é feita usando a classe **PDO**.
4. Se a conexão for estabelecida com sucesso, são definidos dois atributos da conexão PDO: **PDO::ATTR\_ERRMODE** como **PDO::ERRMODE\_EXCEPTION** para habilitar o lançamento de exceções em caso de erros, e **PDO::ATTR\_EMULATE\_PREPARES** como **false** para desativar a emulação de prepared statements.
5. Em caso de falha na conexão, a exceção **PDOException** é capturada e uma mensagem de erro é exibida na tela.
6. O objeto de conexão PDO é retornado para uso posterior.

Método: **\_\_construct($db)**.

Descrição: O método construtor **\_\_construct()** é responsável por inicializar a classe **OperationsUser** e estabelecer a conexão com o banco de dados.

Parâmetro: **$db** (objeto da classe **Connection**) - uma instância da classe **Connection** que contém o método **getConnection()**.

Retorno: Nenhum.

Etapas do Método:

1. O método **\_\_construct()** recebe o objeto **$db** da classe **Connection** como parâmetro.
2. A propriedade privada **$conn** da classe **OperationsUser** é definida como a conexão obtida chamando o método **getConnection()** do objeto **$db**.
3. Essa conexão será usada para executar operações relacionadas ao banco de dados dentro da classe **OperationsUser**.

Método: **create($postValues)**.Decrição: é responsável por criar um novo registro na tabela de usuários (**$this->table\_name**) do banco de dados. Ele recebe um array (**$postValues**) contendo os valores necessários para criar o novo registro, como o email, nome completo, nome de usuário e senha.

Parâmetro: **$postValues** (array) - um array contendo os valores necessários para criar o novo registro na tabela de usuários.

Retorno: Retorna **true** se o registro for criado com sucesso e **false** em caso de falha na execução da consulta SQL.

Curitiba – PR

2023

Etapas do método:

1. O método **create($postValues)** recebe o array **$postValues** que contém os valores necessários para criar o novo registro.
2. Os valores necessários são extraídos do array **$postValues** e atribuídos a variáveis locais: **$EMAIL**, **$FULL\_NAME**, **$USERNAME** e **$PASSWORD**.
3. É criada uma consulta SQL de inserção usando a estrutura de comando **INSERT INTO**, informando a tabela (**$this->table\_name**) e as colunas (**EMAIL**, **FULL\_NAME**, **USERNAME**, **PASSWORD**).
4. Uma instrução preparada (**$stmt**) é preparada a partir da conexão PDO (**$this->conn**) usando a consulta SQL criada.
5. Os valores são vinculados aos parâmetros da instrução preparada usando o método **bindParam()** para evitar injeção de SQL. Cada valor é vinculado a um parâmetro numerado correspondente na consulta SQL.
6. A instrução preparada é executada usando o método **execute()**.
7. Se a execução da instrução preparada for bem-sucedida, ou seja, o registro for criado com sucesso, o método retorna **true**. Caso contrário, retorna **false**.

Método: **login()** Descrição: Este método verifica se foram enviados valores para os campos de email, usuário e senha através do método POST. Em seguida, ele valida os valores fornecidos, executando uma consulta SQL para verificar se há um registro correspondente na tabela de usuários. Se a autenticação for bem-sucedida, uma sessão é iniciada e o ID do usuário é armazenado na variável de sessão.

Modificador de Acesso: **public**

Retorno: Nenhum (vazio).

Etapas do Método:

1. O método **login()** verifica se as variáveis **$\_POST["EMAIL"]**, **$\_POST["USERNAME"]** e **$\_POST["PASSWORD"]** estão definidas, ou seja, se os campos de email, usuário e senha foram enviados no formulário de login.
2. Se algum campo estiver vazio (**strlen($\_POST["EMAIL"]) == 0**, **strlen($\_POST["USERNAME"]) == 0** ou **strlen($\_POST["PASSWORD"]) == 0**), uma mensagem de alerta é exibida usando JavaScript para informar ao usuário que os campos devem ser preenchidos.
3. Caso contrário, se todos os campos tiverem sido preenchidos, as variáveis **$USERNAME**, **$PASSWORD** e **$EMAIL** são inicializadas com os valores fornecidos no formulário.
4. É criada uma consulta SQL para selecionar todos os campos da tabela de usuários onde o nome de usuário (**USERNAME**), a senha (**PASSWORD**) e o email (**EMAIL**) correspondam aos valores fornecidos no formulário.
5. Uma instrução preparada (**$stmt**) é preparada a partir da conexão PDO (**$this->conn**) usando a consulta SQL criada.

Curitiba – PR

2023

1. Os valores do formulário são vinculados aos parâmetros da instrução preparada usando o método **bindParam()** para evitar injeção de SQL. Cada valor é vinculado a um parâmetro numerado correspondente na consulta SQL.
2. A instrução preparada é executada usando o método **execute()**.
3. O resultado da consulta é obtido usando o método **fetch(PDO::FETCH\_ASSOC)**, que retorna um array associativo contendo os campos do registro encontrado na tabela de usuários.
4. Se houver um resultado, ou seja, a autenticação for bem-sucedida, uma sessão é iniciada usando **session\_start()**, e o ID do usuário (**$ID**) é armazenado na variável de sessão **$\_SESSION['ID']**.
5. Caso contrário, se não houver resultado da consulta ou as informações de login estiverem incorretas, uma mensagem de alerta é exibida para informar ao usuário que houve uma falha ao fazer o login.

Método: **logout()** Descrição: Este método realiza as seguintes etapas para efetuar o logout do usuário:

Modificador de Acesso: **function** (não pertence a uma classe específica)

Retorno: Nenhum (vazio).

Etapas do Método:

1. Inicia a sessão usando **session\_start()** para ter acesso às variáveis de sessão.
2. Remove todas as variáveis de sessão existentes usando **session\_unset()**.
3. Destroi a sessão atual usando **session\_destroy()**, removendo todas as informações relacionadas à sessão do servidor.
4. Define os cabeçalhos HTTP para evitar o cache da página usando as seguintes declarações:

**header("Cache-Control: no-cache, no-store, must-revalidate")**

**header("Pragma: no-cache")**

**header("Expires: 0")**

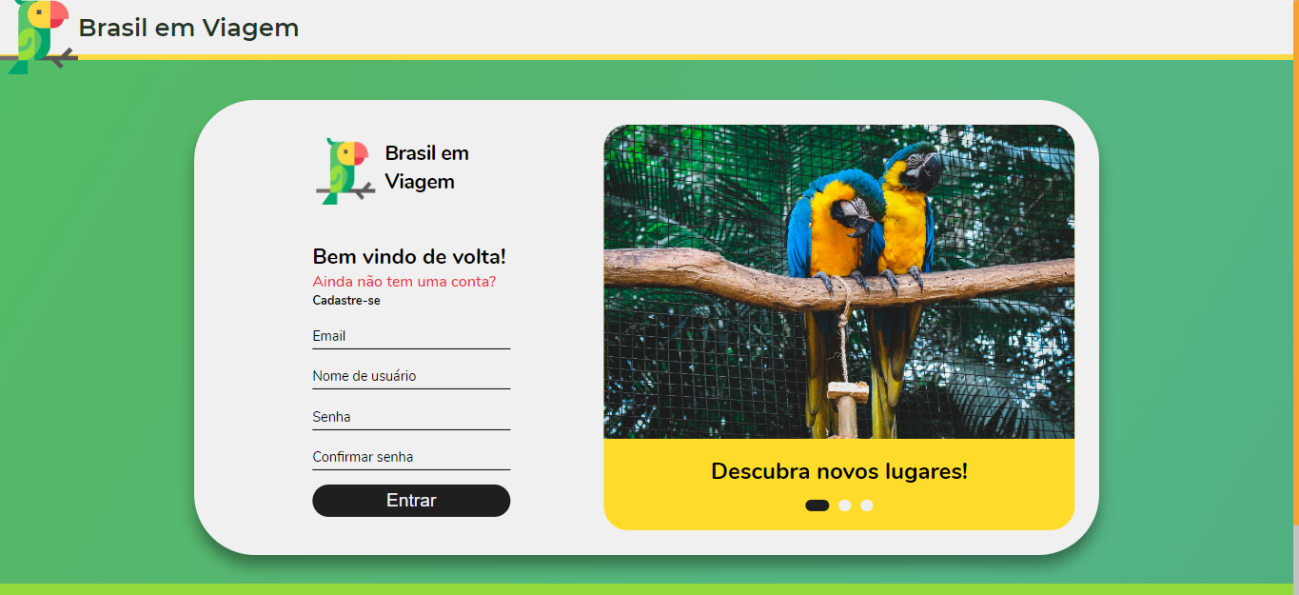
Essas instruções garantem que a página não seja armazenada em cache pelo navegador e seja recarregada do servidor quando o usuário acessá-la novamente.

1. Redireciona o usuário para a página inicial de login/cadastro usando **header("location: ../index.php")**. O caminho **../index.php** pode ser ajustado conforme a estrutura do seu projeto.
2. Encerra o script PHP usando **exit()**, garantindo que nenhuma outra instrução seja executada após o redirecionamento.

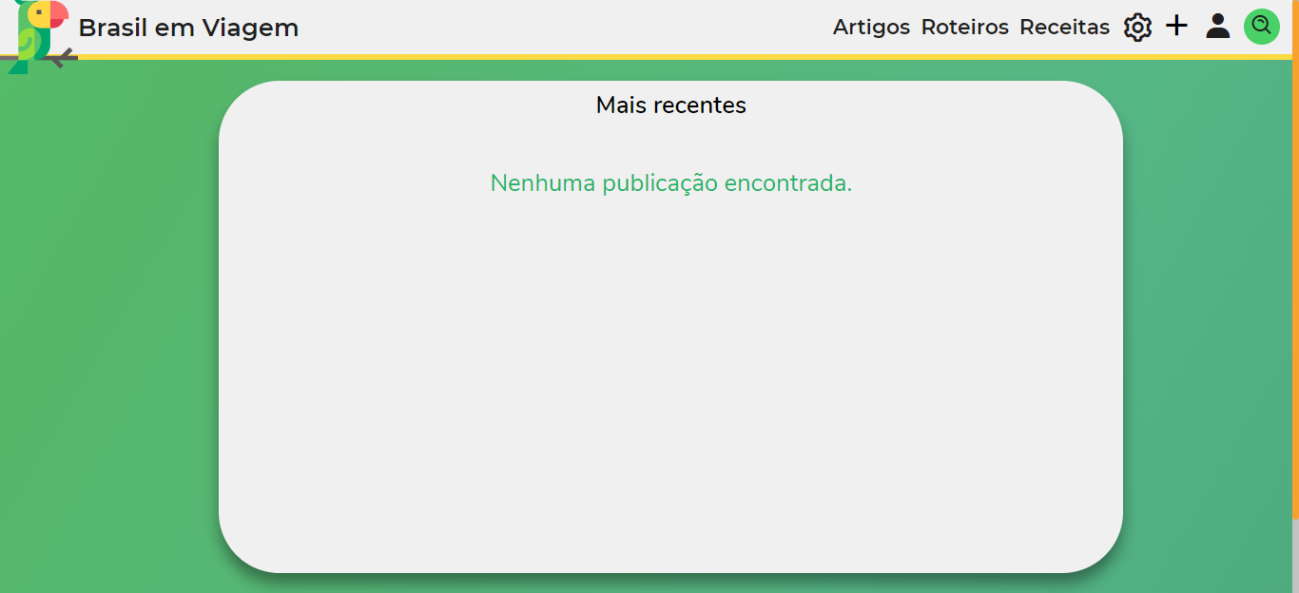
Curitiba – PR

2023

# 4 TELAS DO SISTEMA



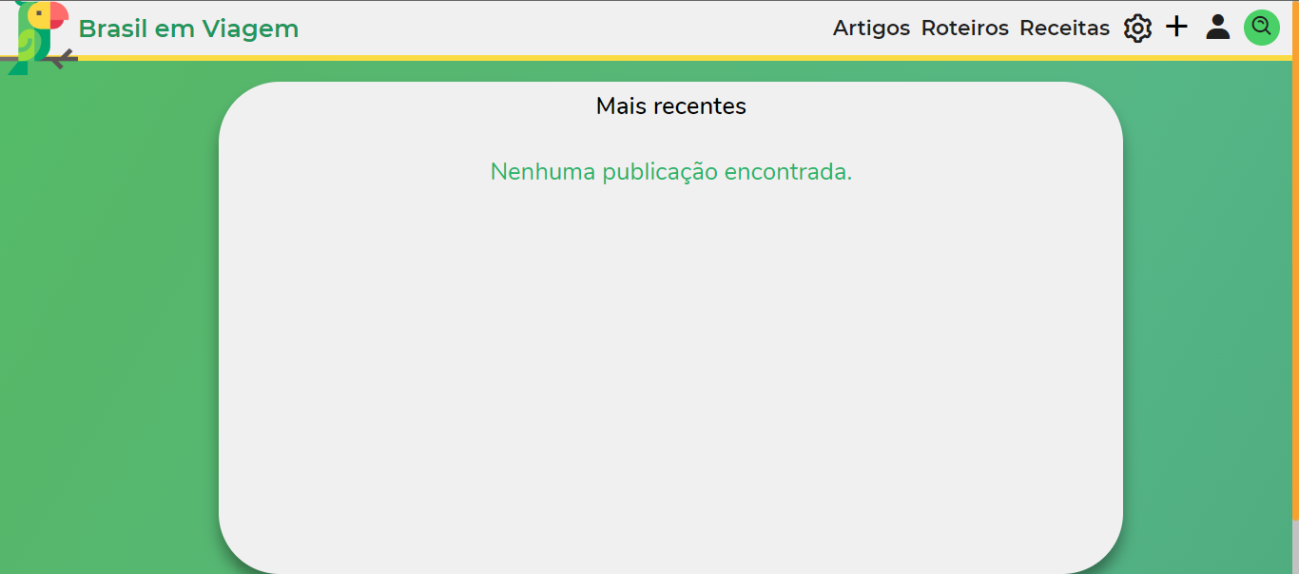
index.php



home.php

Curitiba – PR

2023



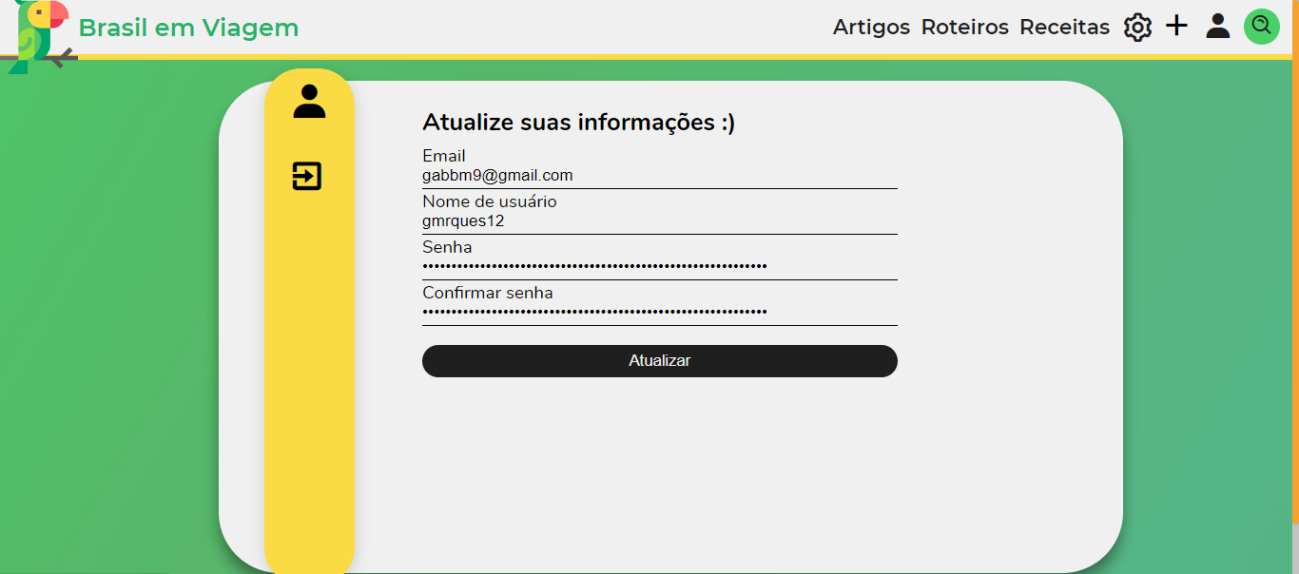
roadmap.php



gastronomy.php

Curitiba – PR

2023



settings.php



profile.php

# 5 DIAGRAMAS

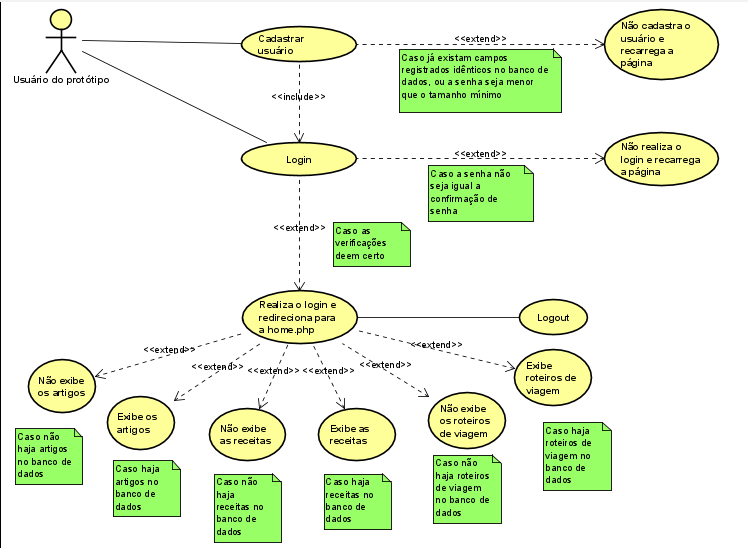


Diagrama de casos de uso do protótipo, não considera as partes não concluídas.

# 6 CONCLUSÃO

O protótipo atual demonstra uma navegação fácil e uma usabilidade intuitiva, proporcionando aos usuários uma experiência agradável. Embora esteja em fase de desenvolvimento, o projeto almeja alcançar uma plataforma totalmente funcional e interativa, permitindo interações entre os usuários, como comentários, compartilhamento nas redes sociais e contribuições nos artigos existentes.

É válido mencionar que existem outras funcionalidades (métodos) e mais telas que não foram abordadas nesta documentação justamente por não terem ficado 100% funcionais a tempo suficiente da entrega.

Curitiba – PR

2023