

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET
CAMPUS GUARAPUAVA

MURILO MAÇANEIRO SCHROEDER

SAMP: SISTEMA DE AMPARO AO PROGRAMADOR

TRABALHO DE CONCLUSÃO DE CURSO DO CURSO SUPERIOR EM
TECNOLOGIA EM SISTEMAS PARA INTERNET

GUARAPUAVA
2019

MURILO MAÇANEIRO SCHROEDER

SAMP: SISTEMA DE AMPARO AO PROGRAMADOR

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso 1, do Curso Superior de Tecnologia em Sistemas para Internet – TSI – da Universidade Tecnológica Federal do Paraná – UTFPR – Campus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Guilherme da Costa Silva

Coorientador: Prof. Dr. Luciano Ogiboski

GUARAPUAVA
2019

ATA DE DEFESA DE MONOGRAFIA TRABALHO DE CONCLUSÃO DE CURSO

No dia **02 de Dezembro de 2019**, às 14:00 horas, em sessão pública nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de defesa de Trabalho de Conclusão de Curso intitulada: "**SAMP: Sistema de Amparo ao Programador**" do(a) acadêmico(a) **Murilo Macaneiro Schroeder** sob orientação do(a) professor(a) **Me. Guilherme da Costa Silva** do curso de Tecnologia em Sistemas para Internet.

Orientação	
Membro	Nome
Orientador	Me. Guilherme da Costa Silva
Coorientador 1	Dr. Luciano Ogiboski

Banca Avaliadora	
Membro	Nome
Orientador	Me. Guilherme da Costa Silva
Avaliador 1	Dr. Roni Fabio Banaszewski
Avaliador 2	Dr. Paulo Henrique Soares

Situação
Aprovado

Guarapuava, 02 de Dezembro de 2019.



Documento assinado eletronicamente por **Guilherme da Costa Silva, Orientador**, em 02/12/2019, às 14:54, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por **Roni Fabio Banaszewski, Avaliador**, em 02/12/2019, às 14:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por **Paulo Henrique Soares, Avaliador**, em 02/12/2019, às 14:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.



Documento assinado eletronicamente por **Murilo Macaneiro Schroeder, Acadêmico**, em 02/12/2019, às 14:56, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site

<https://tcc.tsi.pro.br/autenticidade/documentos>, informando o código verificador **1575309139**.

RESUMO

SCHROEDER, Murilo M. SAMP: Sistema de Amparo ao Programador. 45 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

Quando o objetivo é programar, provavelmente o uso de algum tipo de framework é indispensável e quando utilizado este apresenta uma grande quantidade de classes e métodos muitas vezes desconhecidos por quem o utiliza. Então, para se sanar essas dúvidas a melhor forma é procurar pela documentação API do framework, mas para uma programação que utilize diversos frameworks, a pesquisa por esses documentos pode ser complicada pela quantidade de ferramentas diferentes para cada framework. Isso pode ocorrer não somente na utilização de diversos frameworks, mas também no desenvolvimento de um projeto que utilize diversas linguagens de programação. Desta forma, surge a necessidade de se utilizar uma ferramenta que disponibilize várias documentações API e facilite a pesquisa por essas informações em um único sistema. O propósito deste trabalho é exatamente esse, possibilitando também a disponibilização de documentação elaboradas pelos próprios usuários.

Palavras-Chave: Sistemas de computação. Aplicações Web. Programação para Internet. PHP (Linguagem de programação de computador).

ABSTRACT

SCHROEDER, Murilo M. SAMP: Sistema de Amparo ao Programador. 45 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2019.

When the goal is to program, probably the use of some kind of framework is indispensable and when used this presents a great number of classes and methods often unknown by those who use it. So, in order to solve these doubts the best way is to search for the documentation API of the framework, but for a programming that uses several frameworks, the search for these documents can be complicated by the number of different tools for each framework. This can occur not only using several frameworks, but also in the development of a project that uses several programming languages. In this way rises the need to use a tool that provides several API documentations and facilitate the search for this information in a single system. The purpose of this work is exactly this, making possible the availability of documentation prepared by the users themselves.

Keywords: Computer systems. Web applications. Internet programming. PHP (Computer program language).

LISTA DE FIGURAS

Figura 1 – Website W3Schools.....	12
Figura 2 – Website Ruby on Rails API.....	13
Figura 3 – Website W3Schools.....	14
Figura 4 – Diagrama de casos de uso.....	21
Figura 5 – Protótipo da página de cadastro de projetos.....	22
Figura 6 – Protótipo da página inicial.....	23
Figura 7 – Protótipo da página de projeto RESTful.....	24
Figura 8 – Protótipo da página de projeto não RESTful.....	25
Figura 9 – Protótipo da página de classe de projeto não RESTful.....	26
Figura 10 – Protótipo da página de funções de classe de projeto não RESTful.....	27
Figura 11 – Modelo do banco de dados.....	28
Figura 12 – Página Inicial.....	29
Figura 13 – Sidebar.....	30
Figura 14 – Navbar.....	30
Figura 15 – Página de Projeto RESTful.....	31
Figura 16 – Página de Projeto Não RESTful.....	32
Figura 17 – Página de Classe.....	32
Figura 18 – Página de Função.....	33
Figura 19 – Área de Entrada de Usuário.....	33
Figura 20 – Área de Cadastro de Usuário.....	34
Figura 21 – Sidebar do Usuário.....	35
Figura 22 – Página da Conta do Usuário.....	35
Figura 23 – Página dos Projetos do Usuário.....	36
Figura 24 – Página de Criação de Projeto.....	37
Figura 25 – Página de Edição de Projeto.....	38
Figura 26 – Página Administrativa.....	39
Figura 27 – Página de Listagem de Usuários.....	39
Figura 28 – Página de Edição de Usuário.....	40
Figura 29 – Página de Listagem de Projetos.....	41
Figura 30 – Página da Conta do Administrador.....	41

LISTA DE SIGLAS

API	Application Programming Interfaces
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
PHP	Hypertext Preprocessor
URL	Uniform Resource Locator
SGBDR	Sistema Gerenciador de Banco de Dados Relacional
RDBMS	Relational Database Management System

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVOS	10
1.1.1 OBJETIVO GERAL	10
1.1.2 OBJETIVOS ESPECÍFICOS	10
2 RESENHA LITERÁRIA	11
2.1 TRABALHOS RELACIONADOS	11
2.1.1 W3SCHOOLS	11
2.1.2 RUBY ON RAILS API	12
2.1.3 PHP MANUAL	13
2.1.4 COMPARATIVOS	14
2.2 FUNDAMENTAÇÃO TEÓRICA	15
2.2.1 HTML	15
2.2.2 JAVASCRIPT	15
2.2.3 CSS	16
2.2.4 PHP	16
2.2.5 LARAVEL	16
2.2.6 SQL	17
2.2.7 RESTFUL	17
3 METODOLOGIA	18
3.1 ESTUDO DE TECNOLOGIAS	18
3.2 LEVANTAMENTO DE REQUISITOS	18
3.3 PROJETO DO SISTEMA	18
3.4 DESENVOLVIMENTO	18
3.5 TESTES	19
4 DESENVOLVIMENTO	20
4.1 REQUISITOS	20
4.1.1 REQUISITOS FUNCIONAIS	20
4.2 DIAGRAMAS DE CASOS DE USO	20
4.3 PROTÓTIPOS DE TELAS	21
4.4 MODELO DE BANCO DE DADOS	28
4.5 DESENVOLVIMENTO DO SISTEMA	29
4.5.1 ÁREA PÚBLICA	29

4.5.2 ÁREA DO USUÁRIO	34
4.5.3 ÁREA DO ADMINISTRADOR	38
5 CONSIDERAÇÕES FINAIS	42
5.1 TRABALHOS FUTUROS	42
REFERÊNCIAS	43

1 INTRODUÇÃO

Por que o programador precisa de ajuda? A maior parte dos códigos criados por programadores são complexos necessitando desta forma a utilização de recursos fornecidos pela própria linguagem de programação ou por outras fontes. Esses recursos podem ajudar tanto iniciantes, criando uma base de conhecimento para problemas futuros, quanto programadores mais experientes, acrescentando ou corrigindo conteúdos já estudados pelo indivíduo.

Atualmente o processo de busca por ajuda em relação a uma determinada linguagem de programação não é tão fácil quanto deveria ser. Este processo de busca por ajuda pode ser feito de diferentes maneiras, como a pesquisa em sites, onde é possível obter exercícios, exemplos, dicas e ideias de códigos de programação compartilhados por outros programadores. Outra forma de ajuda é procurar pela documentação oficial da linguagem de programação específica, estes são chamados de documentações Application Programming Interfaces (APIs).

Documentações APIs possuem diversos significados e podem ser utilizados em diferentes áreas da computação. Geralmente o uso de uma API está relacionada a frameworks e bibliotecas de uma linguagem de programação, mas seu uso pode variar dependendo da linguagem utilizada. Em uma linguagem orientada a objetos, como o Java, e em outras linguagens comumente utilizadas como HTML, CSS e Javascript, a API provém uma lista de classes e interfaces com respectivos métodos e descrições, assim como, detalhes de erros, exceções e diversas outras informações. (MAALEJ e ROBILLARD, 2013; ORACLE, 2018).

O uso da API nos frameworks está relacionado a um conjunto de várias classes e métodos de bibliotecas, onde são descritos com uma codificação reduzida e simplista para facilitar o entendimento. Entretanto, essa ajuda está demasiadamente espalhada através da internet. Levando em consideração que uma grande parte das linguagens de programação estão interligadas, a quantidade de ferramentas que seriam necessárias para se obter ajuda em cada linguagem seria grande, causando, desta forma, desconforto e uma significativa diminuição de desempenho do programador.

Para corrigir essa queda no desempenho foi desenvolvido o Sistema de Amparo ao

Programador (SAMP), onde é possível a visualização de documentações disponibilizadas pelos usuários do sistema. Possibilitando a busca por documentações de diversas linguagens de programação em único sistema.

No sistema desenvolvido é possível realizar a pesquisa por documentações APIs de projetos disponibilizadas nele, esses por sua vez podem ser cadastrados no sistema por usuários que anteriormente foram registrados no mesmo. Desta forma o sistema resolve o problema das informações dispersas através da internet, juntando-as em um único sistema e consequentemente aumentando o desempenho do programador.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

Desenvolver um sistema web de ajuda a programadores, com disponibilização de documentações APIs de diversas linguagens de programação em um único sistema, fornecidas tanto pelo sistema quanto pelos próprios usuários.

1.1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho são:

- Implementar módulo para registro de documentação API através de arquivos de documentação fornecidos pelos usuários;
- Implementar módulo para busca e visualização de conteúdo através de palavras-chave;
- Implementar módulo para cadastro e gerenciamento de usuários;
- Implementar módulo de comentários nas páginas dos métodos das APIs e classificação dos mesmos;

2 RESENHA LITERÁRIA

2.1 TRABALHOS RELACIONADOS

O SAMP atende a um grupo de usuários, provendo serviços de geração e disponibilização de documentações APIs. A seguir são apresentadas algumas ferramentas com funcionalidades similares aos do SAMP.

2.1.1 W3SCHOOLS

O W3School é uma das maiores ferramentas didáticas para o auxílio e ensino a linguagens de programação, como HTML, CSS e Javascript. Ele possui um grande número de tutoriais, referências e exemplos simplificados para ajudar no entendimento de determinada linguagem. A Figura 1 apresenta a página inicial do W3Schools onde é possível visualizar as categorias de diversas linguagens de programação. (W3SCHOOLS, 2018).

Diferente do sistema desenvolvido ele não possui uma área para cadastro e interações de usuários como por exemplo campos de comentários. É possível notar também que são apresentados apenas documentações de projetos mais conhecidos como Bootstrap, AngularJS, entre outros.

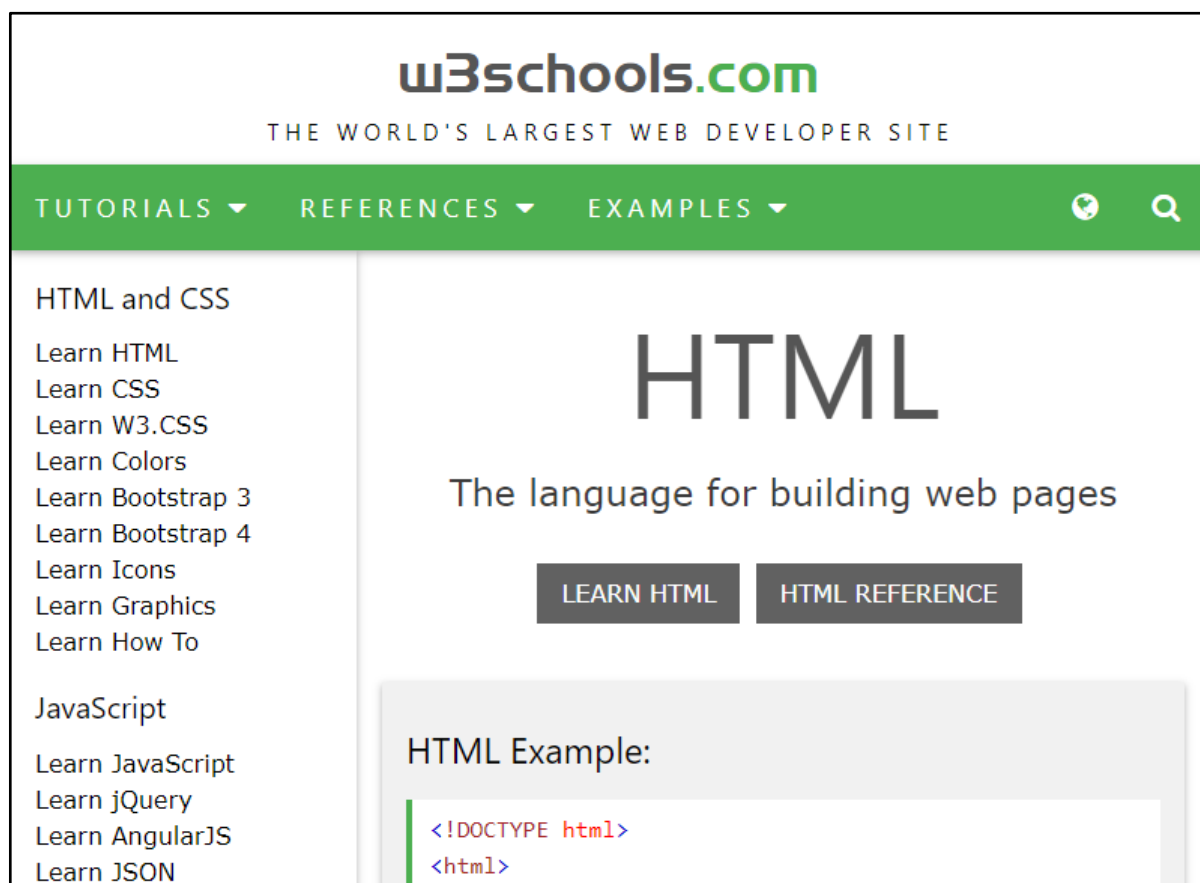


Figura 1: W3Schools Website.

Fonte: W3Schools.

2.1.2 RUBY ON RAILS API

Ruby on Rails API é outra ferramenta utilizada para o auxílio na programação, neste caso é específica para o *framework* Ruby on Rails, essa documentação apresenta os módulos, classes e métodos do *framework*, além de uma grande quantidade de exemplos em cada seção como mostra a Figura 2. (RUBY ON RAILS, 2018).

Search

- files
- AbstractController
- ActionController
 - TestCase < ActiveSupport::TestCase
- ActionDispatch
 - IntegrationTest < ActiveSupport::TestCase
- ActionMailer
- ActionView
- ActiveJob
- ActiveModel
- ActiveRecord
- ActiveSupport
 - TestCase < Minitest::Test
 - Array < Object**
 - Benchmark
 - BigDecimal < Object
 - Class < Object
 - Date < Object
 - DateAndTime
 - DateTime < Object

Ruby on Rails 4.2.10

Class

Array < Object

activesupport/lib/active_support/core_ext/array/access.rb
 activesupport/lib/active_support/core_ext/array/conversions.rb
 activesupport/lib/active_support/core_ext/array/extract_options.rb
 activesupport/lib/active_support/core_ext/array/grouping.rb
 activesupport/lib/active_support/core_ext/array/prepend_and_append.rb
 activesupport/lib/active_support/core_ext/array/wrap.rb
 activesupport/lib/active_support/core_ext/enumerable.rb
 activesupport/lib/active_support/core_ext/object/blank.rb
 activesupport/lib/active_support/core_ext/object/deep_dup.rb
 activesupport/lib/active_support/core_ext/object/json.rb
 activesupport/lib/active_support/core_ext/object/to_query.rb

Methods

- D deep_dup
- E extract_options!
- F fifth, forty_two, fourth, from
- I in_groups, in_groups_of

Figura 2: Ruby on Rails API.

Fonte: Ruby on Rails.

2.1.3 PHP MANUAL

Assim como o Ruby on Rails API, o PHP Manual possui inúmeros exemplos e como o próprio nome já diz é específico para a linguagem PHP. Pode-se visualizar sua página principal na Figura 3. Ele também disponibiliza uma área em cada página para que usuários deixem seus exemplos com o intuito de ajudar e/ou facilitar a programação assim como o SAMP. (PHP MANUAL, 2018).

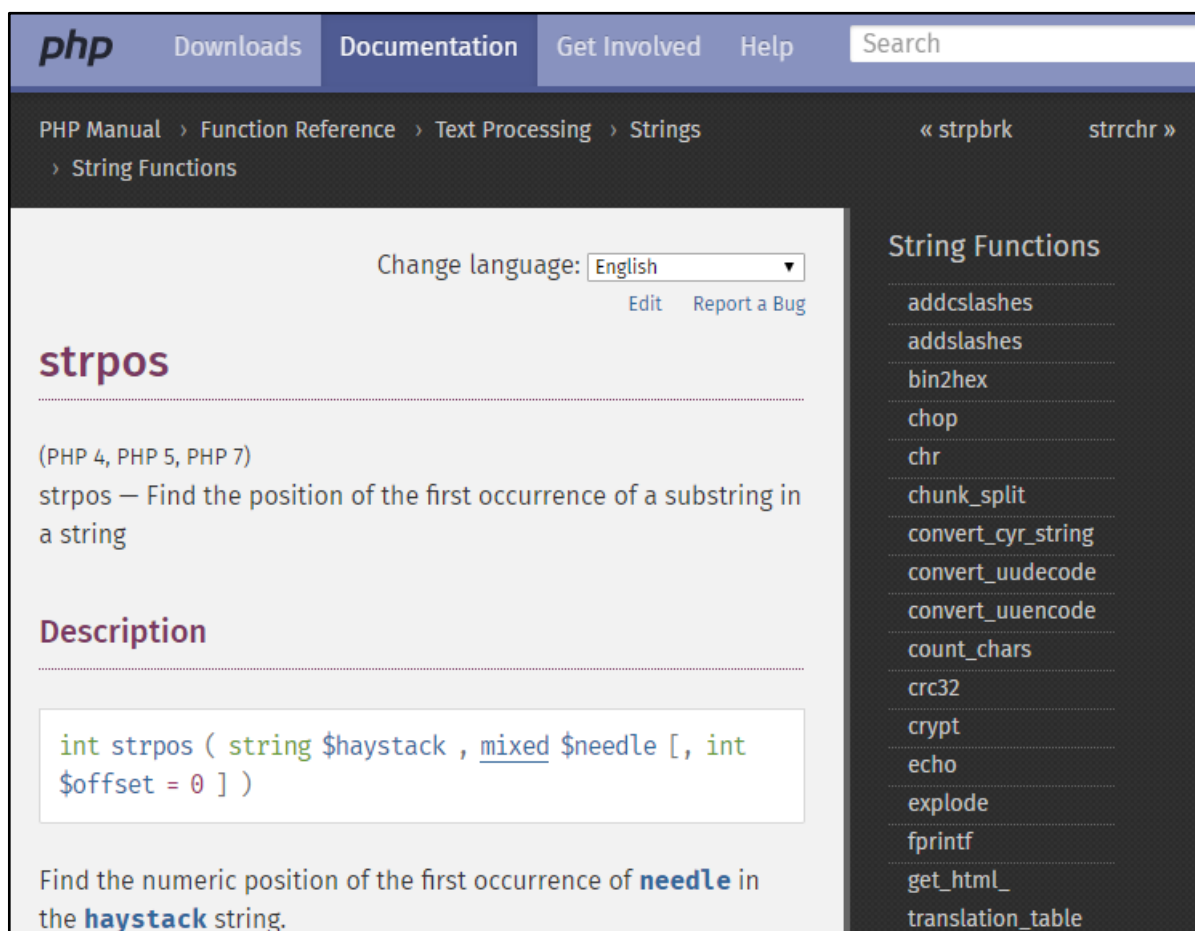


Figura 3: PHP Manual.

Fonte: PHP.

2.1.4 COMPARATIVOS

No Quadro 1 é feita uma comparação entre as ferramentas citadas anteriormente e o sistema desenvolvido, com o foco nas principais funcionalidades identificadas durante o processo de estudo. Note que a principal diferença está na disponibilização de APIs de usuários cadastrados através de arquivos de documentação gerados por ferramentas como ApiDoc e documentation.js.

Quadro 1 - Diferencial Tecnológico

Ferramentas	Web	Mobile	Disponibilização de APIs de usuários cadastrados	Pesquisa por palavras-chave	Variedade em linguagens de programação	Comentários e exemplos de usuários nas páginas das APIs
W3Schools	X	X		X	X	
Ruby on Rails API	X	X		X		
PHP Manual	X	X		X		X
SAMP	X	X	X	X	X	X

Fonte: Autoria própria

2.2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção são descritas as tecnologias utilizadas para a realização do TCC.

2.2.1 HTML

HTML conhecida como Linguagem de Marcação de HiperTexto é utilizada para definir conteúdos em uma página web através de blocos de elementos. Esses blocos são definidos por marcações ou *markup* e são usadas para envolver conteúdos como textos, imagens e outros. (MDN WEB DOCS, 2018).

Os HiperTextos são referentes aos links que fazem a ligação entre uma página e outra, podendo ser internas ou externas ao website. (MDN WEB DOCS, 2018).

2.2.2 JAVASCRIPT

JavaScript é uma linguagem de programação script de alto nível e dinâmica podendo desta forma alterar elementos do código enquanto está sendo executada. Atualmente a maior

parte dos navegadores como Firefox, Chrome ou Safari. Essa linguagem é bem flexível e expressiva possibilitando a criação de aplicações robustas. (JONES, 2017).

Essa linguagem permite implementar funcionalidades complexas para páginas web. Com ela é possível gerenciar conteúdos dinamicamente como por exemplo mapas interativos, animações gráficas 2D e 3D, etc. (MDN WEB DOCS, 2018).

2.2.3 CSS

CSS em português, Folha de Estilos em Cascata, é uma linguagem utilizada para estilizar conteúdos e elementos apresentados em um documento HTML. (MDN WEB DOCS, 2018).

O CSS pode controlar tanto os estilos e designs da página, exemplos são cores, posicionamento, fontes entre outros. (W3SCHOOLS, 2018).

2.2.4 PHP

PHP é uma linguagem de desenvolvimento web podendo ser utilizada juntamente com HTML. Diferente de JavaScript ou CSS onde o código é executado no cliente o PHP é executado no servidor onde é gerado o HTML que é disponibilizado para o cliente. (PHP, 2019).

2.2.5 LARAVEL

Laravel é um *framework* que estende a linguagem de programação PHP. O Laravel serve para a criação de websites através da combinação do PHP, HTML, CSS e JavaScript. É considerado o *back-end* de plataforma de desenvolvimento de aplicação web ou *server-side* de um website. Seu objetivo é facilitar a criação de aplicações web com a utilização de novos métodos e padrões como por exemplo o Model-View-Controller. (LARAVEL, 2019).

2.2.6 SQL

Structured Query Language (SQL) ou Linguagem de Consulta Estruturada é uma linguagem baseada em modelo relacional para manipulação e consulta de dados salvos em um banco de dados relacionais e tornou-se um padrão em 1986. (MICROSOFT, 2018; OPPEL, 2015).

Existem diversas definições para banco de dados, mas a maior parte das fontes de conhecimento sobre o assunto assumem que banco de dados é uma coleção de dados organizados em um formato estruturado definido como *metadata* ou em português metadados que descreve essa estrutura, ou seja, *metadata* são os dados sobre os dados armazenados no banco de dados. (OPPEL, 2015).

Alguns Sistema Gerenciador de Banco de Dados Relacional (SGBDR), do inglês *Relational Database Management System* (RDBMS) que utilizam SQL são Oracle, MySQL, SQL Server e diversos outros. (OPPEL, 2015).

2.2.7 RESTFUL

RESTful são serviços web que possuem o estilo arquitetural REST proposto por Roy Fielding. Essa arquitetura provê *uniform interface semantics* ou semântica de interface uniforme onde essencialmente são implementadas interfaces para Criar, Recuperar, Atualizar e Deletar recursos de serviços web através de *URLs*. (W3, 2018).

3 METODOLOGIA

Nesta seção descreve-se o processo metodológico para o desenvolvimento do sistema. Os passos metodológicos são descritos a seguir:

3.1 ESTUDO DE TECNOLOGIAS

Nesta etapa foram estudados a linguagem PHP e o *framework* Laravel que foram utilizados no desenvolvimento deste TCC, assim como também, o estudo de ferramentas de geração de documentação API. Foi estudado a estrutura de arquivos, normalmente do tipo Json, gerados pelas ferramentas para que então as informações fossem interpretadas pelo sistema.

3.2 LEVANTAMENTO DE REQUISITOS

Através do estudo realizado anteriormente, foram levantados os requisitos funcionais, permitindo então compreender a necessidade das funcionalidades desenvolvidas neste TCC.

3.3 PROJETO DO SISTEMA

Após o levantamento dos requisitos foi realizado a modelagem do banco de dados e prototipação das principais telas do sistema. Esses itens eram essenciais durante o desenvolvimento do sistema. A prototipação foi feita utilizando draw.io, ferramenta gratuita que possui um grande número de elementos para criação de protótipos.

3.4 DESENVOLVIMENTO

Durante esta etapa foi realizada a implementação do sistema em forma de código de acordo com os resultados obtidos anteriormente. Para o desenvolvimento foi utilizado como base a linguagem PHP juntamente com o *framework* Laravel e outros *frameworks*.

3.5 TESTES

Após o desenvolvimento foi feito os testes e verificações das funcionalidades implementadas para que estivessem de acordo com o que foi definido durante a etapa de levantamento de requisitos e também a verificação para que o sistema não dispusesse de falhas de segurança como *sql injection*, um tipo de invasão que pode ser perigoso para os dados do sistema e dos usuários. E caso fosse necessário uma refatoração do código seria feita. Essa verificação foi realizada através de testes manuais.

4 DESENVOLVIMENTO

4.1 REQUISITOS

Através dos estudos realizados neste trabalho foram levantados os requisitos funcionais a seguir.

4.1.1 REQUISITOS FUNCIONAIS

Nesta seção são apresentados os requisitos funcionais:

- Cadastro de usuários;
- Gerenciamento de dados de usuários cadastrados;
- Cadastro de documentação de projeto;
- Através da linguagem e ferramenta selecionadas no cadastro de projeto o sistema deverá distinguir se é do tipo RESTful ou não.
- Pesquisar documentação de projeto por: nome, tipo (RESTful ou não RESTful) e linguagem;
- Criação de comentário em função de projetos do tipo não RESTful;
- Gerar relatórios na área administrativa de quantidade de acessos a projetos, classes e funções;
- Votar positivamente ou negativamente em comentários;
- Gerenciamento de usuários e projetos na área administrativa;

4.2 DIAGRAMAS DE CASOS DE USO

A partir dos requisitos funcionais definidos anteriormente foi desenvolvido o diagrama de casos de uso abaixo.

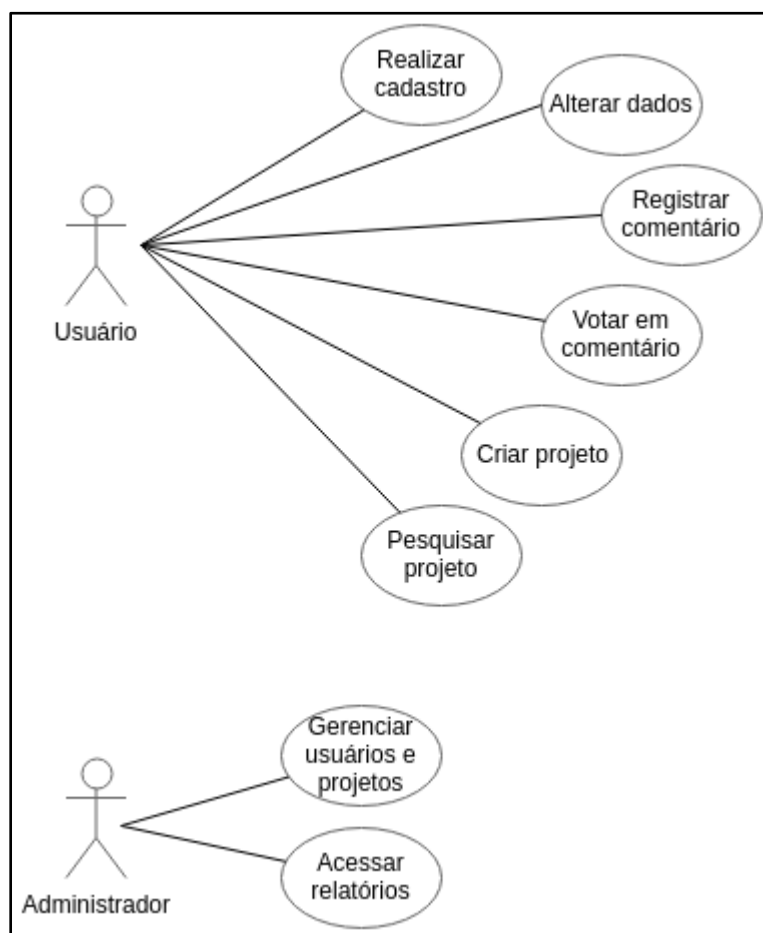


Figura 4: Diagrama de casos de uso

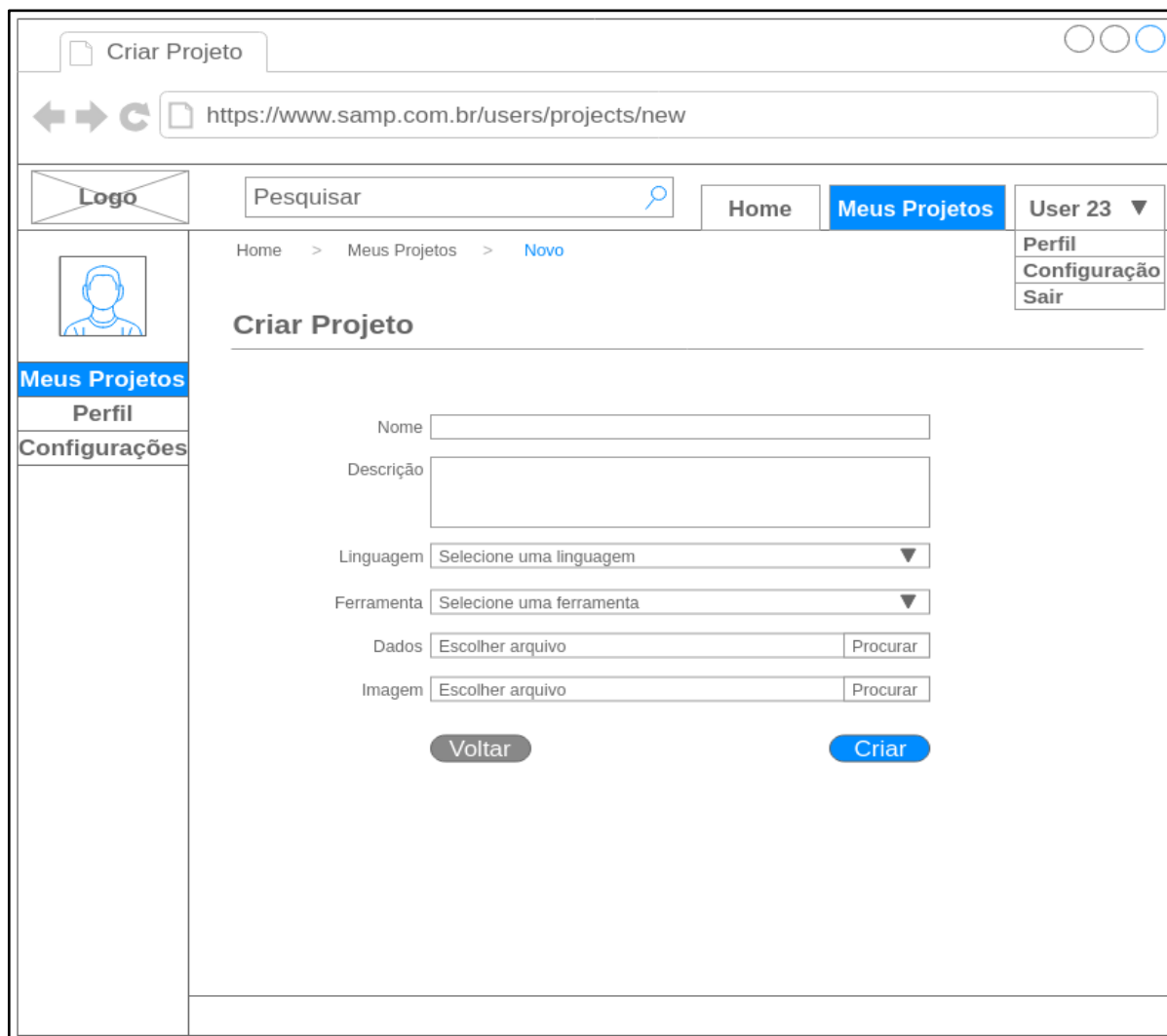
Fonte: Autoria própria

4.3 PROTÓTIPOS DE TELAS

A seguir são apresentados os protótipos das principais páginas desenvolvidas no sistema. Na Figura 5 é apresentada a página onde é possível realizar o cadastro das documentações a partir de um arquivo de dados gerado por uma das ferramentas listadas anteriormente. A página inicial do sistema pode ser observada na Figura 6, onde a mesma apresenta os projetos já cadastrados no sistema e com a possibilidade de realizar pesquisas e filtrar tanto por linguagens quanto por tipo de projeto.

O sistema possui dois tipos de projeto, RESTful e Não RESTful. Na figura 7 apresenta-se em uma única página um projeto RESTful e suas respectivas classes, funções, parâmetros e exemplos. As páginas de um projeto Não RESTful podem ser observadas nas Figuras 8, 9 e 10. Projetos como por exemplo *frameworks*, bibliotecas e outros foram considerados do tipo Não RESTful e foram divididos em página do projeto onde são listadas

as classes, página da classe onde são listadas as funções e, por fim, página da função. Note que a página de uma função apresenta seu respectivos parâmetros, conteúdo, exemplos e comentários onde usuários podem postar seus exemplos de aplicação da função.



O protótipo da página de cadastro de projetos apresenta uma interface web com os seguintes elementos:

- Barra de Navegação:** Contém o botão "Logo", um campo de busca "Pesquisar", e links para "Home" e "Meus Projetos" (destacado em azul).
- Perfil do Usuário:** Exibe "User 23" com um menu suspenso contendo as opções "Perfil", "Configuração" e "Sair".
- Sequência de Navegação:** "Home > Meus Projetos > Novo".
- Formulário "Criar Projeto":**
 - Nome:** Campo de texto único.
 - Descrição:** Campo de texto de maior altura.
 - Linguagem:** Menu suspenso com o texto "Selecione uma linguagem".
 - Ferramenta:** Menu suspenso com o texto "Selecione uma ferramenta".
 - Dados:** Campo de upload com o texto "Escolher arquivo" e um botão "Procurar".
 - Imagem:** Campo de upload com o texto "Escolher arquivo" e um botão "Procurar".
- Botões de Ação:** "Voltar" (cinza) e "Criar" (azul).
- Menu Lateral:** Localizado no canto inferior esquerdo, com opções "Meus Projetos" (destacado em azul), "Perfil" e "Configurações".

Figura 5: Protótipo da página de cadastro de projetos

Fonte: Autoria própria

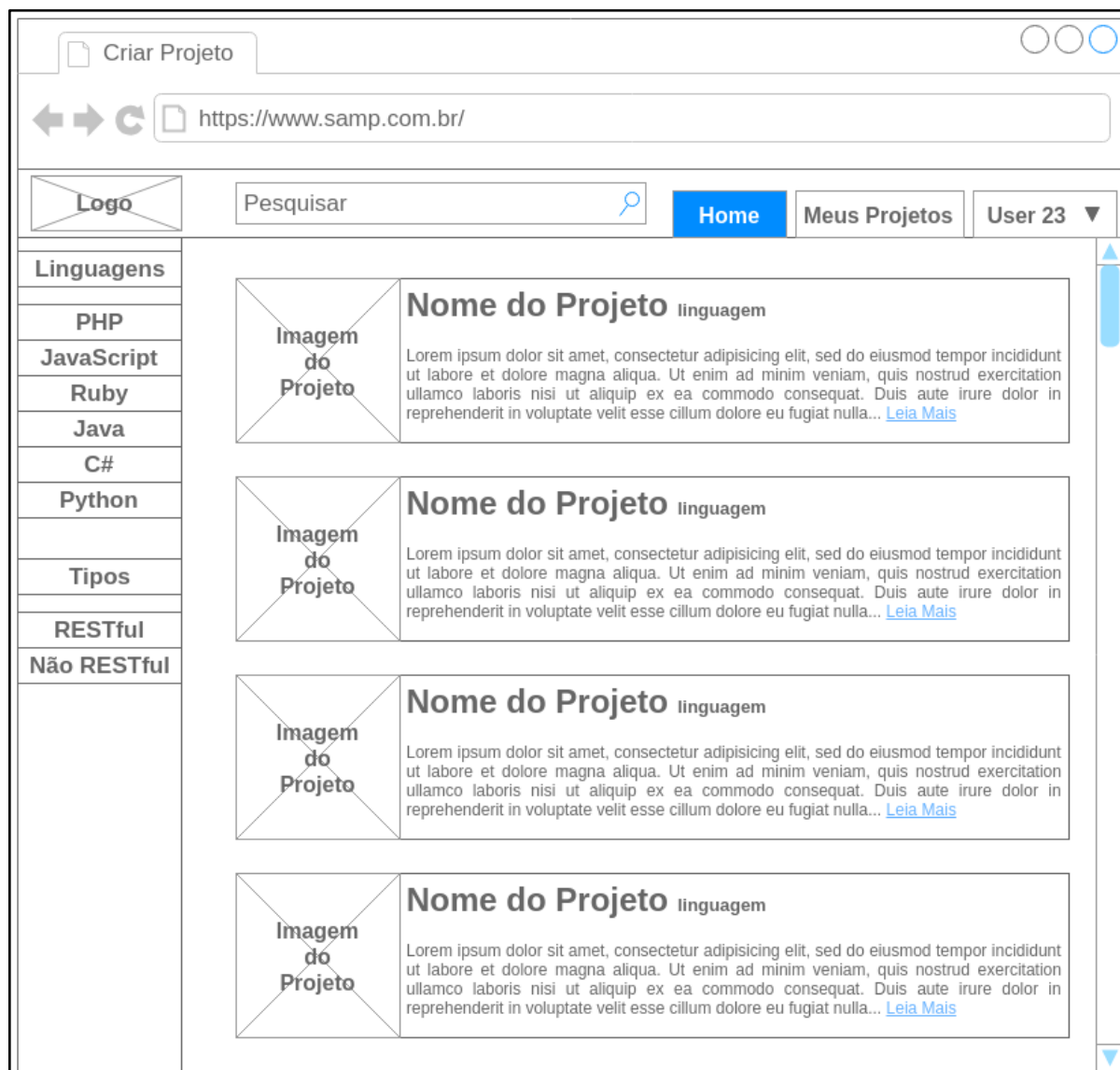


Figura 6: Protótipo da página inicial

Fonte: Autoria própria

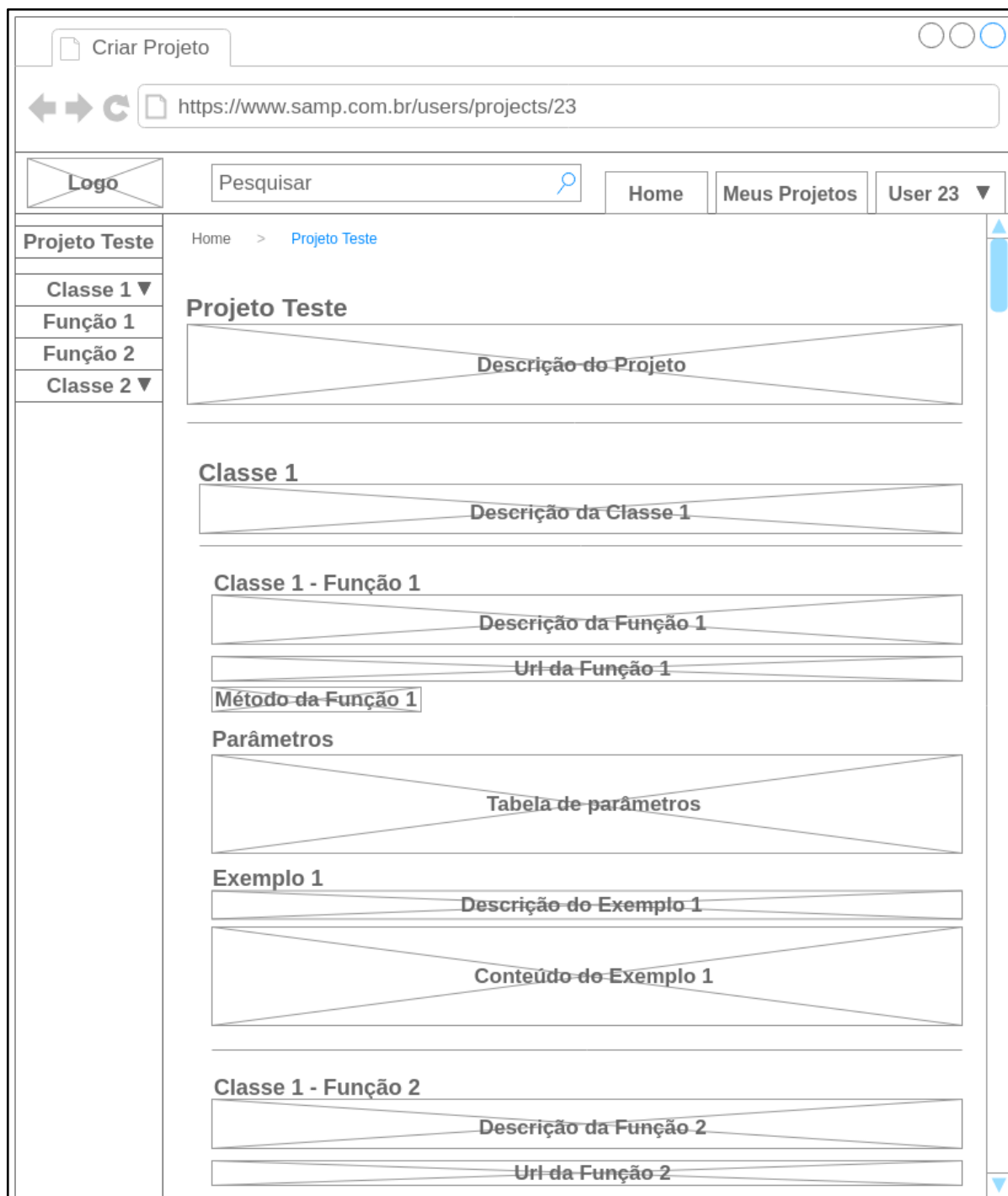


Figura 7: Protótipo da página de projeto RESTful

Fonte: Autoria própria

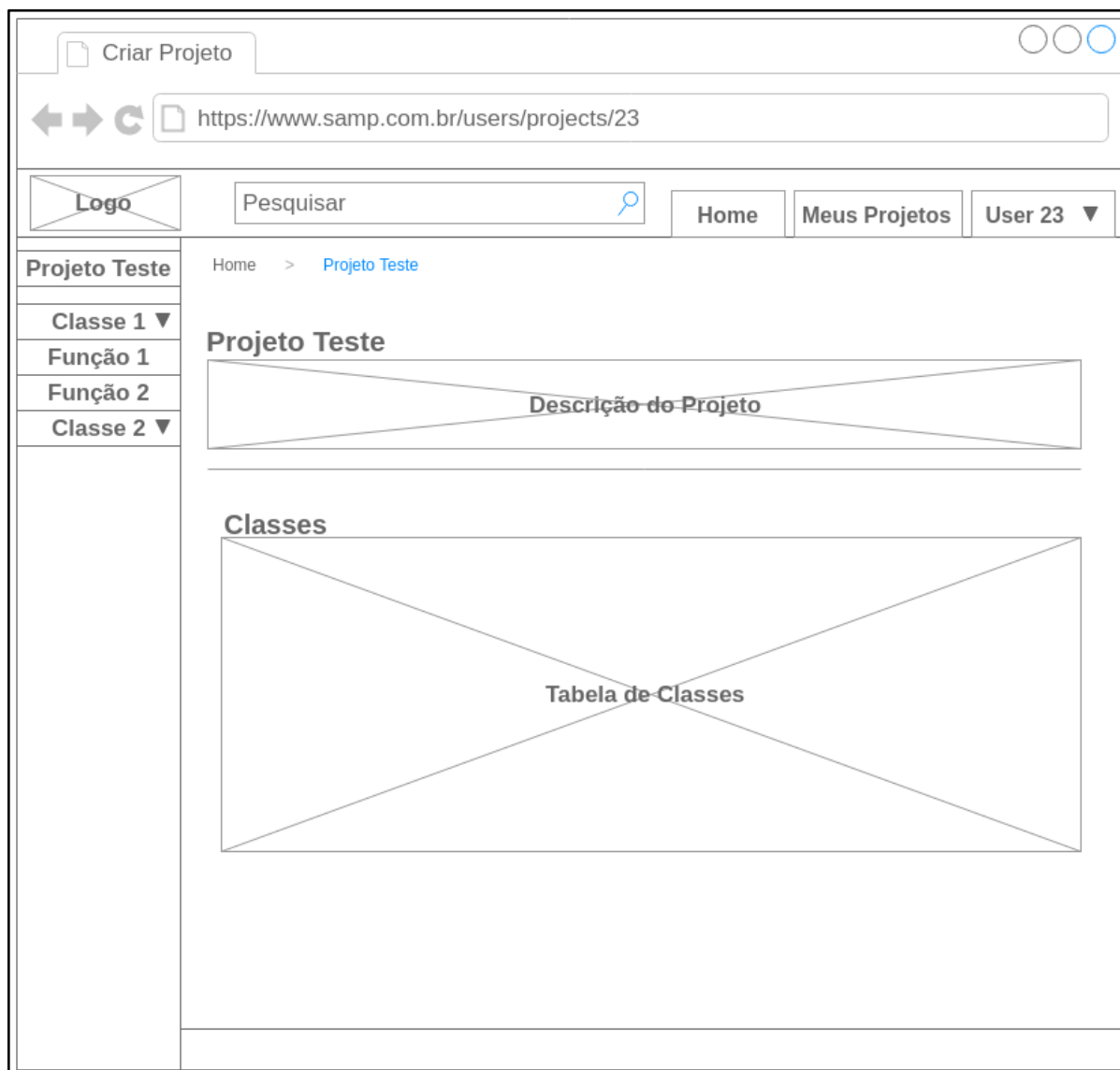


Figura 8: Protótipo da página de projeto não RESTful

Fonte: Autoria própria

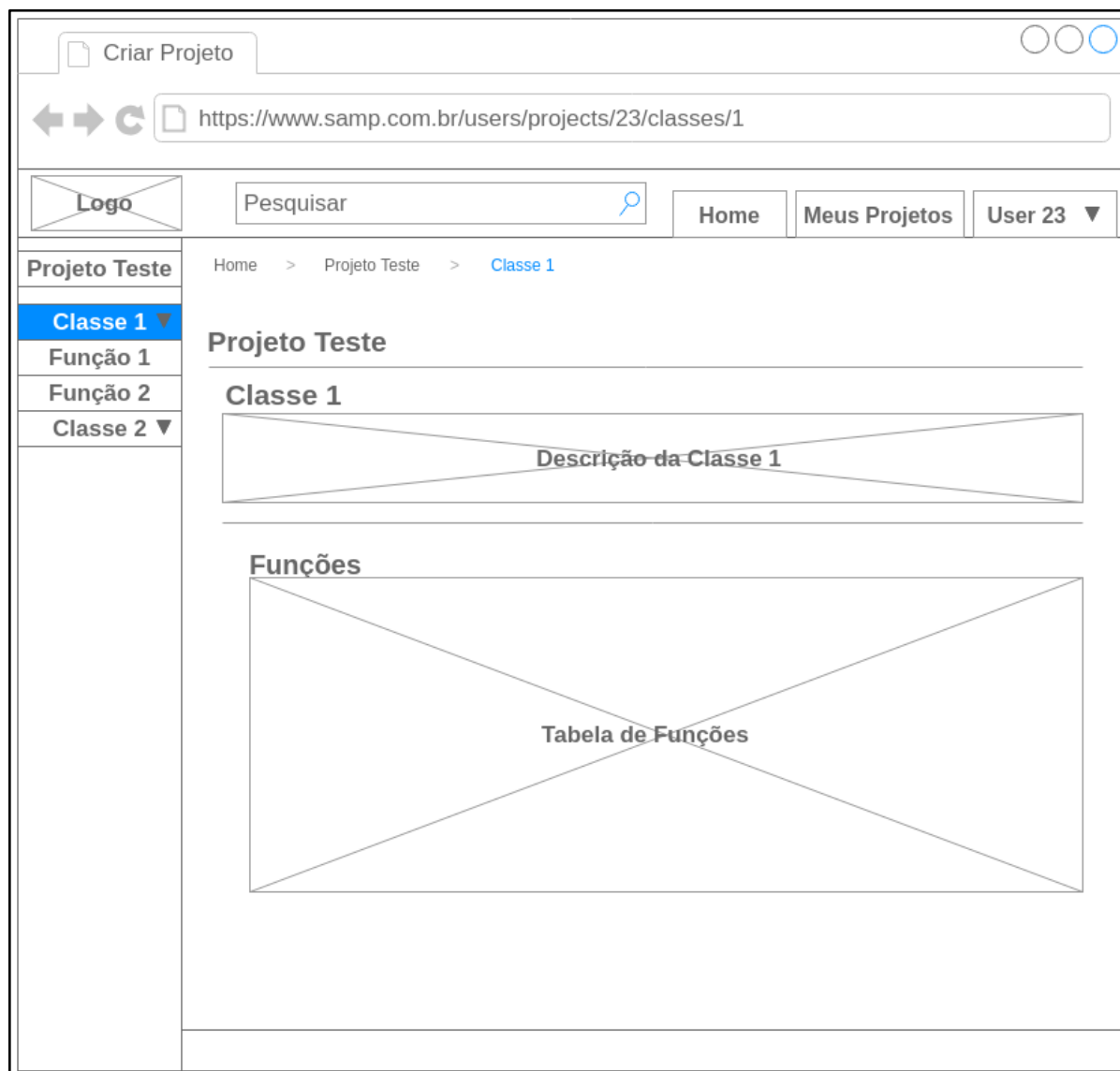


Figura 9: Protótipo da página de classe de projeto não RESTful

Fonte: Autoria própria

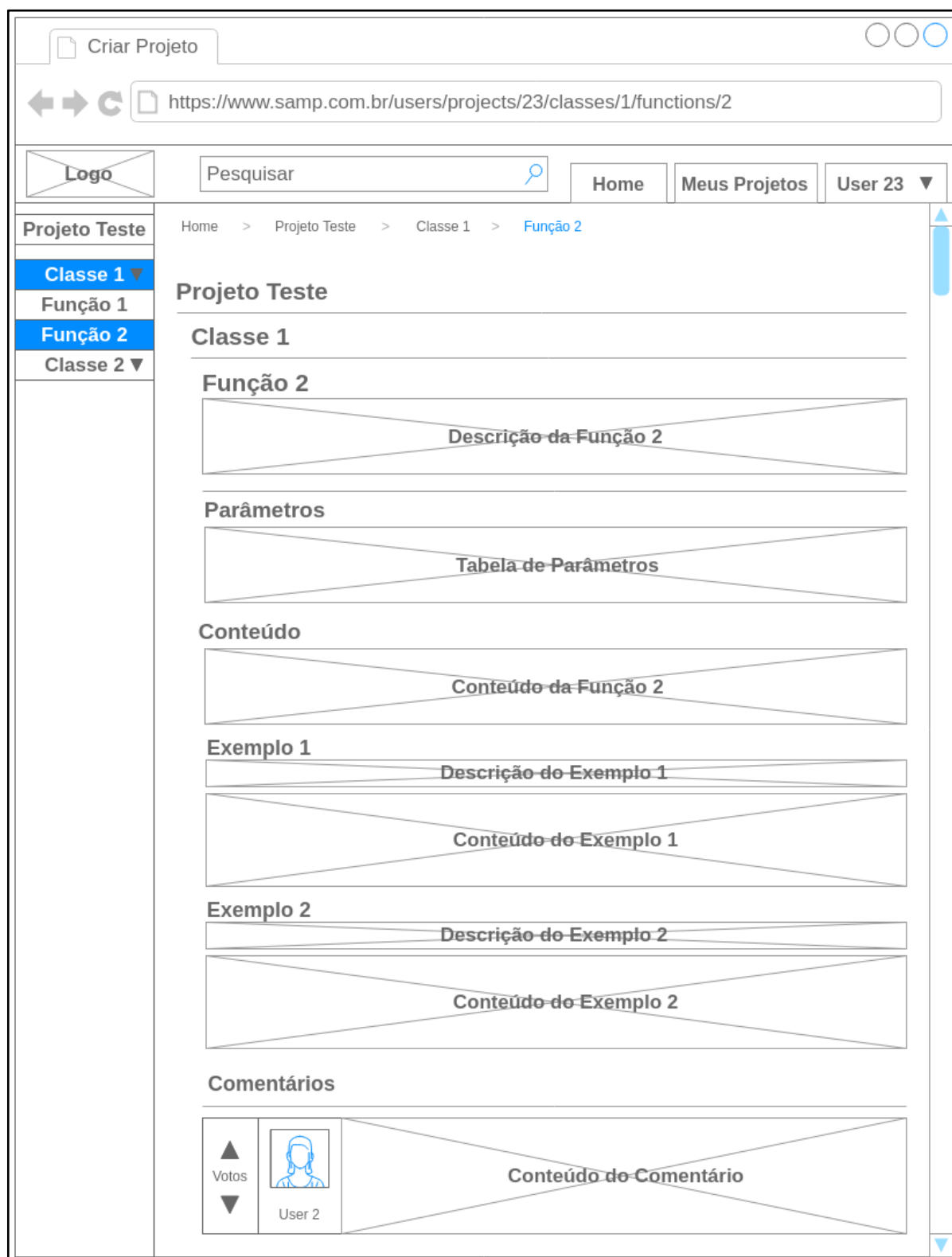


Figura 10: Protótipo da página de funções de classe de projeto não RESTful

Fonte: Autoria própria

4.4 MODELO DE BANCO DE DADOS

A partir dos estudos realizados neste trabalho foram selecionadas algumas ferramentas que apresentam a geração de um arquivo Json. Entre as ferramentas estão ApiDoc e documentation.js. Tendo como base a estrutura dos arquivos gerados pelas ferramentas foi feita a modelagem do banco de dados apresentado na Figura 11.

Informações sobre usuários e administradores são salvos respectivamente nas tabelas *users* e *admins*. Os usuários comuns podem cadastrar projetos a partir da prévia definição de qual linguagem e qual ferramenta foram utilizadas para geração da documentação. Os projetos possuem de forma hierárquica classes, funções, parâmetros, exemplos e comentários.

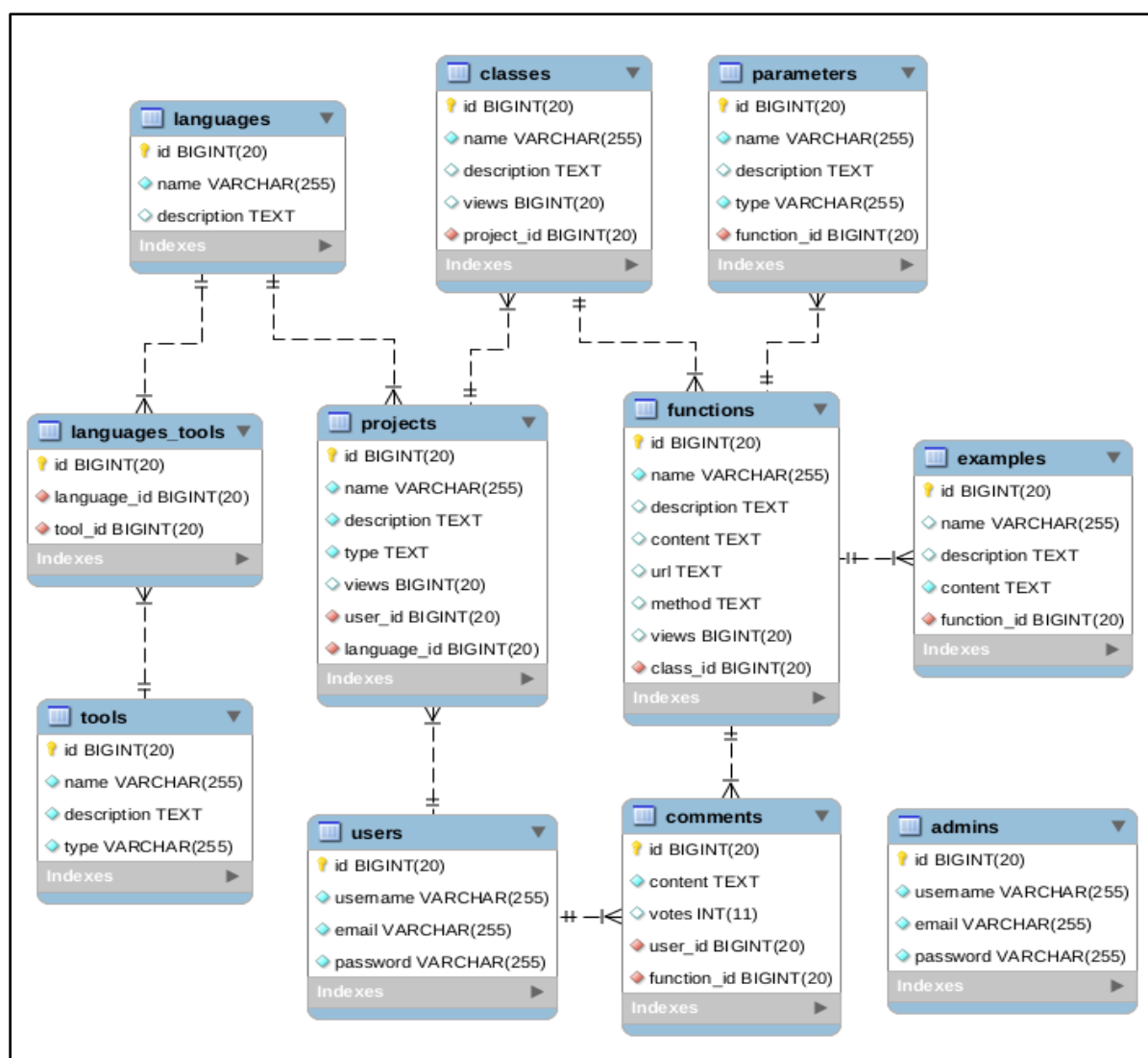


Figura 11: Modelo do banco de dados

Fonte: Autoria própria

4.5 DESENVOLVIMENTO DO SISTEMA

Após a prototipação das telas e modelagem do banco de dados teve início o desenvolvimento do sistema.

4.5.1 ÁREA PÚBLICA

A página inicial do sistema observada na Figura 12 pode ser acessada por visitantes, usuários e administradores. Nela são apresentados os projetos já cadastrados no sistema contendo o nome, descrição e imagem do mesmo.

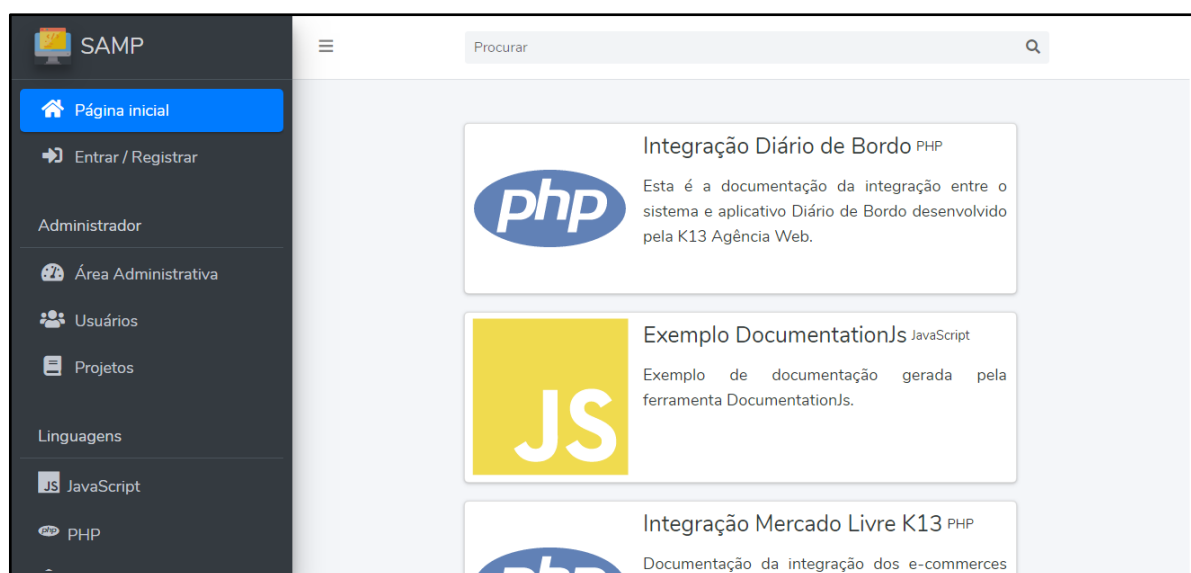


Figura 12: Página Inicial

Fonte: Autoria própria

O *sidebar* apresentado na Figura 13 possui os filtros de busca por projetos a partir da linguagem ou tipo do projeto e também possui um botão para entrada ou cadastro de um usuário. Além disso o *sidebar* dispõe de algumas variações quando na página do projeto e quando administrador estiver logado.

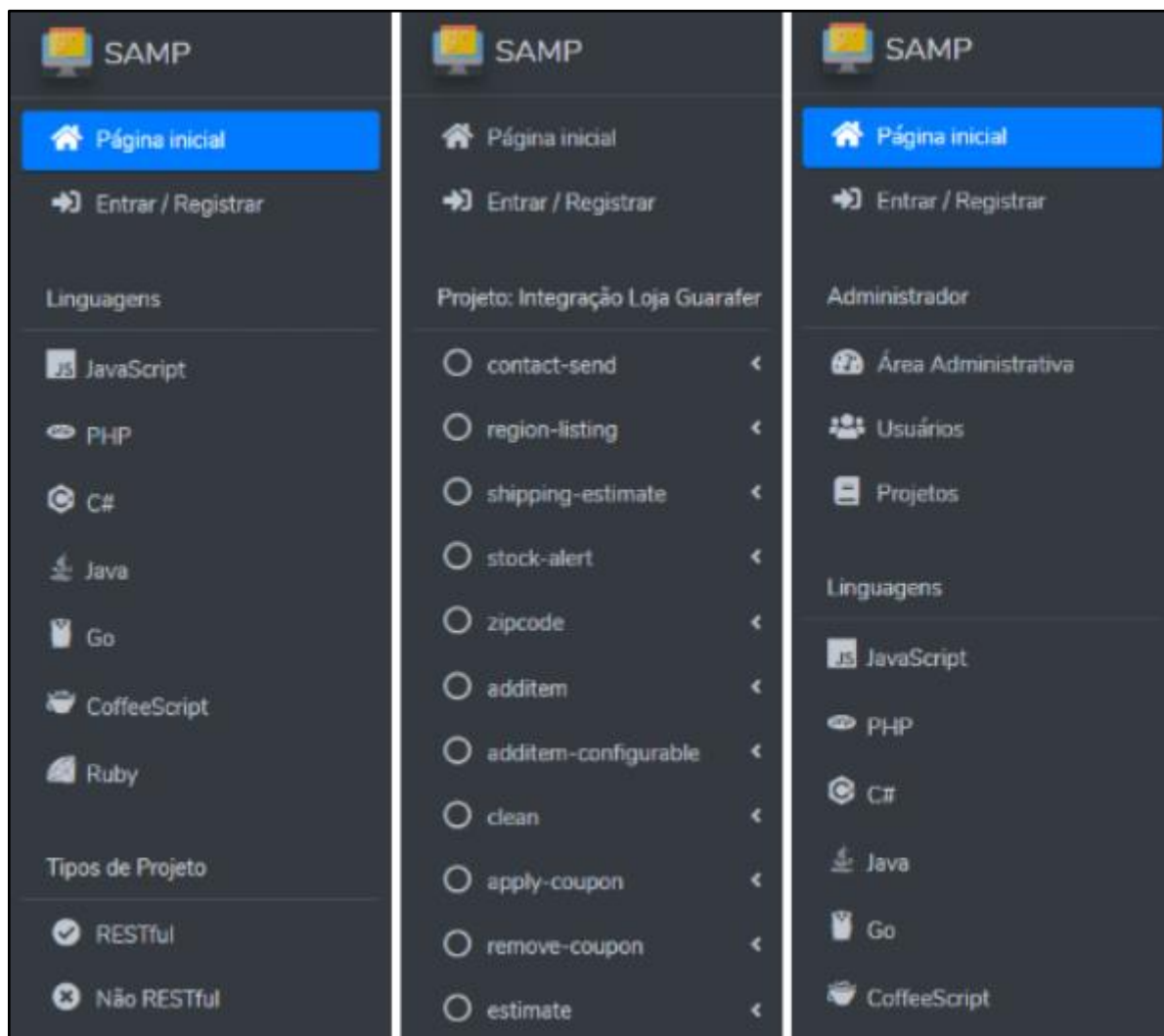


Figura 13: Sidebar

Fonte: Autoria própria

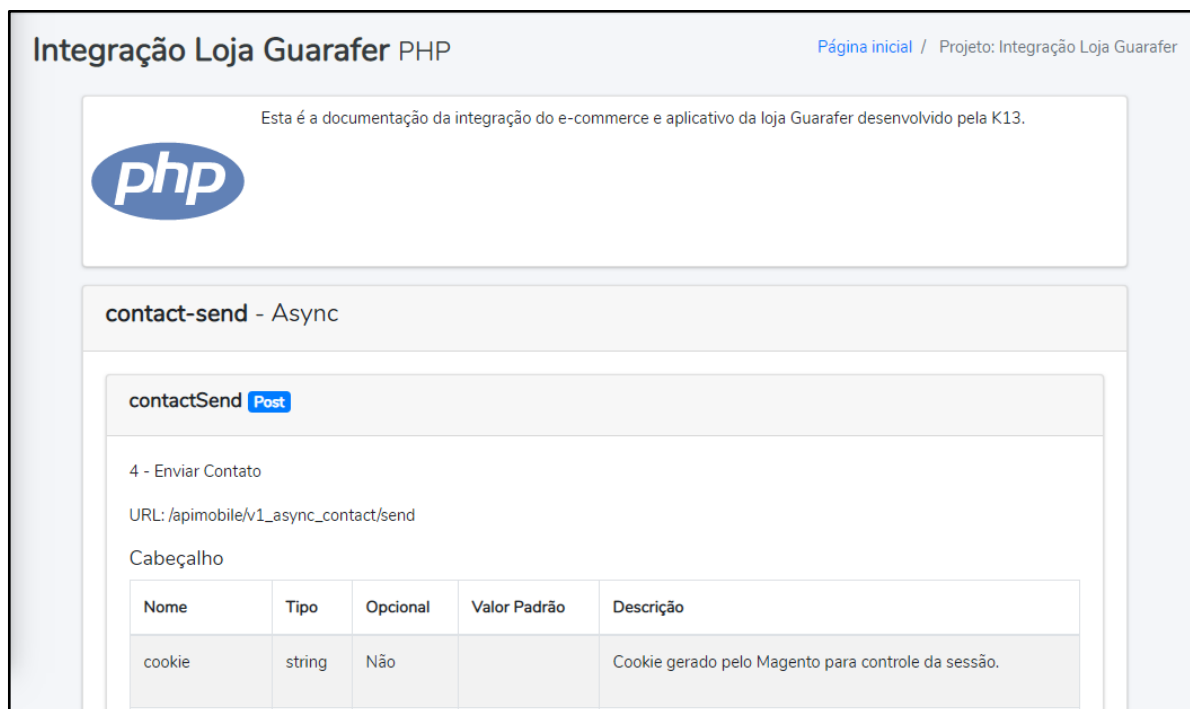
O *navbar* da Figura 14 possui um campo para pesquisa de projetos pelo nome ou descrição do mesmo. E também possui variações quando um usuário ou administrador estiver logado.



Figura 14: Navbar

Fonte: Autoria própria

Nas Figuras 15 e 16 são apresentadas as páginas dos projetos RESTful e Não RESTful respectivamente. Note que para os projetos RESTful todas as informações do mesmo são apresentadas em uma única página, enquanto que projetos Não RESTful as informações são separadas por página, classe é apresentada na Figura 17 e função na Figura 18.



Integração Loja Guarafer PHP [Página inicial](#) / Projeto: Integração Loja Guarafer

Esta é a documentação da integração do e-commerce e aplicativo da loja Guarafer desenvolvido pela K13.

contact-send - Async

contactSend Post

4 - Enviar Contato

URL: /apimobile/v1_async_contact/send

Cabeçalho

Nome	Tipo	Opcional	Valor Padrão	Descrição
cookie	string	Não		Cookie gerado pelo Magento para controle da sessão.

Figura 15: Página de Projeto RESTful

Fonte: Autoria própria

Exemplo DocumentationJs JavaScript [Página inicial](#) / Projeto: Exemplo DocumentationJs

Exemplo de documentação gerada pela ferramenta DocumentationJs.

Classes

Mostrar 5 por página

Nome	Descrição	Opções
Popup	Creates a popup component	Q
Map	Creates a map instance.	Q
GeoJSONSource	Create a GeoJSON data source instance given an options object	Q
Navigation	Creates a navigation control with zoom buttons and a compass	Q

Figura 16: Página de Projeto Não RESTful

Fonte: Autoria própria

Exemplo DocumentationJs JavaScript [Página inicial](#) / [Projeto: Exemplo DocumentationJs](#) / Classe: Popup

Classe: `Popup(Object options, Boolean options.closeButton, Boolean options.closeOnClick)`

Creates a popup component

Parâmetros

Nome	Tipo	Opcional	Descrição
options	Object	Não	
options.closeButton	Boolean	Não	
options.closeOnClick	Boolean	Não	

Figura 17: Página de Classe

Fonte: Autoria própria

Exemplo DocumentationJs

[Página inicial](#) /
 [Projeto: Exemplo DocumentationJs](#) /
 [Classe: Popup](#) /
 Função: setText

JavaScript

Classe: Popup

Função De Instância: setText(string text)

Populate a popup element with text only content

Parâmetros

Nome	Tipo	Opcional	Descrição
text	string	Não	

Retorno

Figura 18: Página de Função

Fonte: Autoria própria

Tanto a área de entrada quanto a de cadastro de usuários podem ser visualizadas nas Figuras 19 e 20 respectivamente.

The image shows a login modal window titled "Entrar" (Login) with a close button (X) in the top right corner. The modal contains the following elements:

- A text input field for the email address, containing "admin@admin.com", with an envelope icon on the right.
- A password input field with masked characters "....." and a lock icon on the right.
- A link "Esqueceu a senha?" (Forgot password?) below the password field.
- A checkbox labeled "Lembre-se de mim" (Remember me).
- A blue button labeled "Entrar" (Login) to the right of the checkbox.
- A separator "- OU -" (or) below the checkbox.
- A green button labeled "Registrar" (Register) at the bottom.

The background is a blurred screenshot of a web page titled "Integração Diário de Bordo PHP" (PHP Daily Log Integration), showing text about a system and API.

Figura 19: Área de Entrada de Usuário

Fonte: Autoria própria

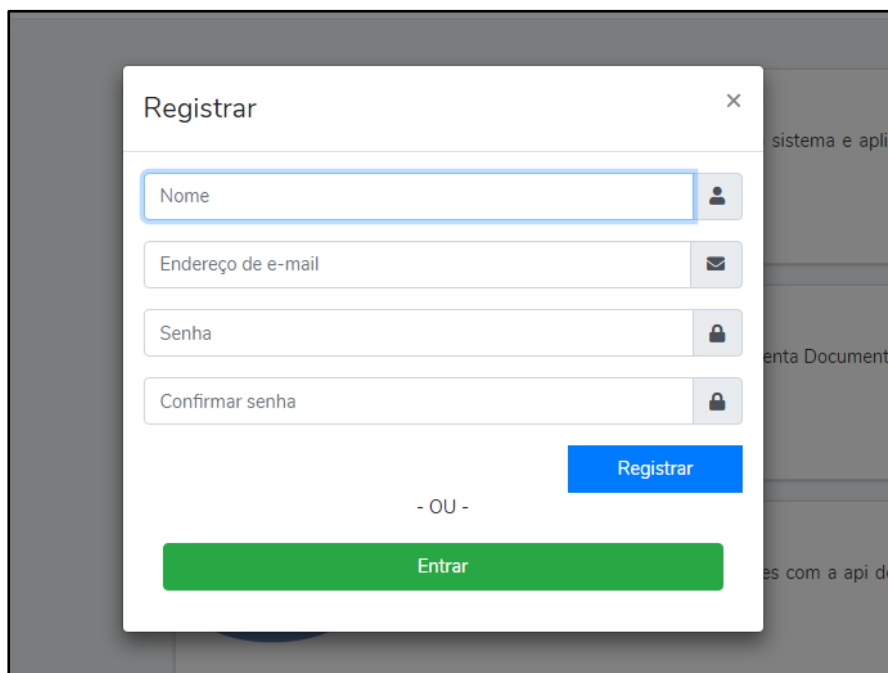
A imagem mostra um modal de registro de usuário com o título "Registrar" e um ícone de fechar (X) no canto superior direito. O modal contém quatro campos de entrada: "Nome" (com ícone de pessoa), "Endereço de e-mail" (com ícone de envelope), "Senha" (com ícone de cadeado) e "Confirmar senha" (com ícone de cadeado). Abaixo dos campos, há um botão azul "Registrar" e o texto "- OU -". No rodapé do modal, há um botão verde "Entrar". O modal está sobreposto a uma interface de usuário desfocada.

Figura 20: Área de Cadastro de Usuário

Fonte: Autoria própria

4.5.2 ÁREA DO USUÁRIO

Após um usuário entrar no sistema um novo *sidebar* apresentado na Figura 21 pode ser acessado através do ícone no lado direito do *navbar*. Nele é possível visualizar botões para acessar os dados da conta, os projetos cadastrados pelo usuário e sair do sistema.

Na página da conta do usuário é possível editar o nome, e-mail e senha do mesmo, essa é visível na Figura 22.

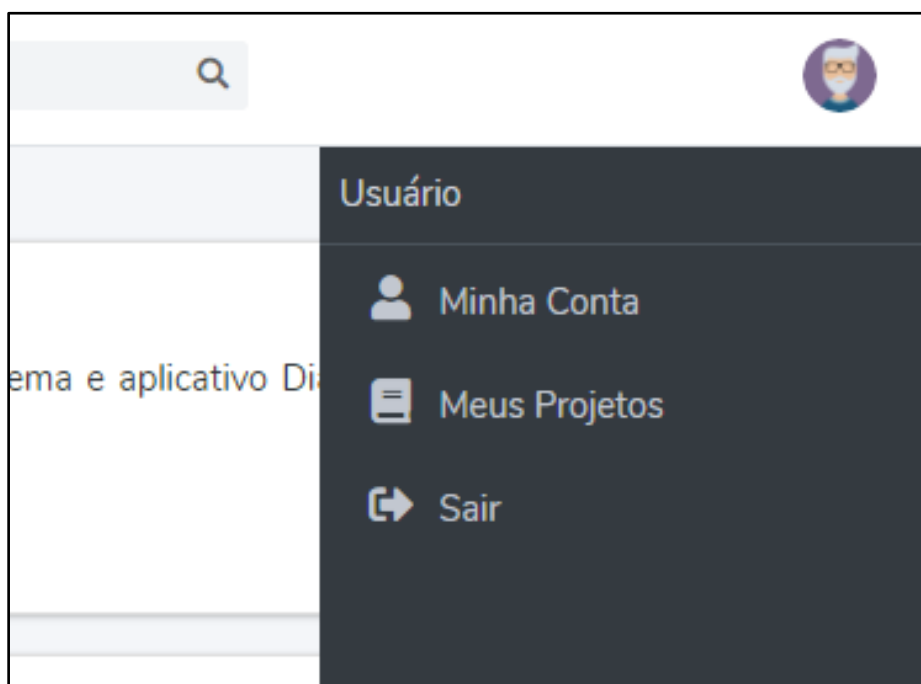


Figura 21: Sidebar do Usuário

Fonte: Autoria própria

Figura 22: Página da Conta do Usuário

Fonte: Autoria própria

Na Figura 23 é apresentada a página dos projetos do usuário onde é possível pesquisar pelo nome do projeto e acessar a criação, edição e visualização de projetos.

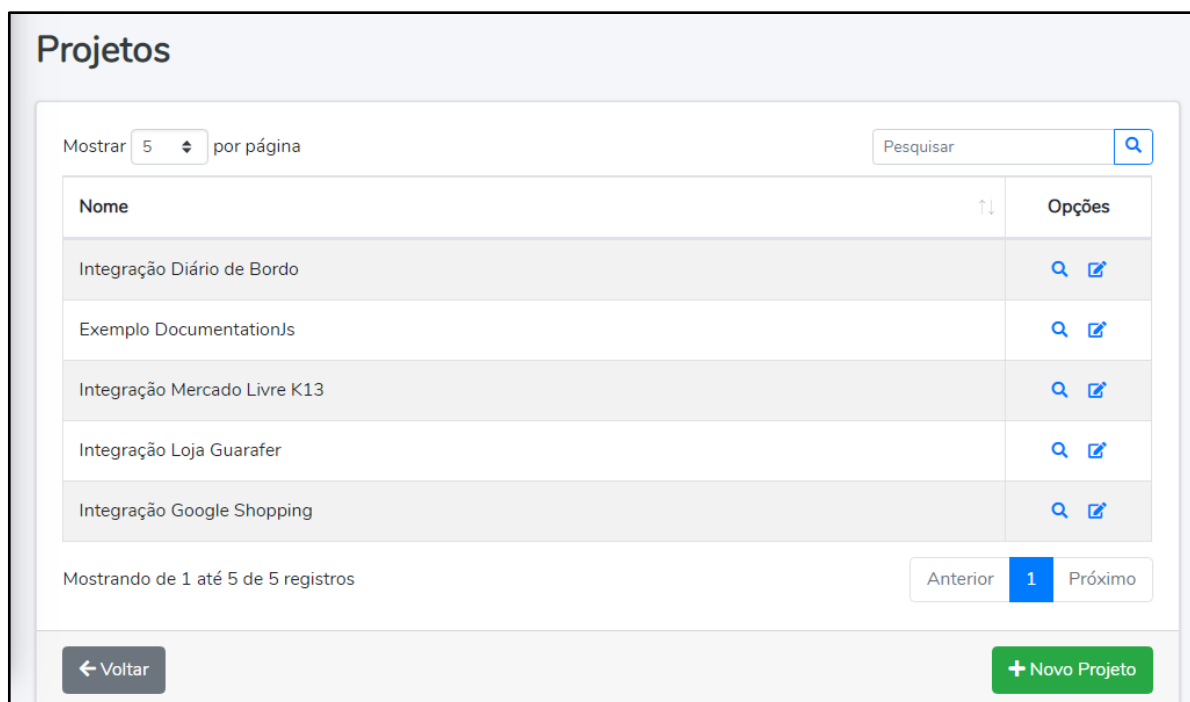


Figura 23: Página dos Projetos do Usuário

Fonte: Autoria própria

Na página de criação de um projeto visível na Figura 24 estão presentes campos para nome, descrição, ferramenta, linguagem, arquivo de dados e imagem. O campo de ferramenta possui as opções ApiDoc e DocumentationJs e cada uma das opções possuem linguagem compatíveis para serem selecionadas.

Diferente da criação, na edição é possível alterar apenas o nome, descrição e imagem do projeto como visto na Figura 25.

Projetos

Projetos / Novo Projeto

Novo Projeto

Nome

Descrição

Ferramenta

Ferramenta

Linguagem

Linguagem

Dados

Selecionar arquivo

Browse

Imagem

Arraste e solte os arquivos aqui...
(ou clique para selecionar o(s) arquivo(s))

Selecionar arquivo...

Procurar...

Voltar

Salvar

Figura 24: Página de Criação de Projeto

Fonte: Autoria própria

Projetos Projetos / Editar Projeto

Editar Projeto

Nome

Integração Diário de Bordo

Descrição

Esta é a documentação da integração entre o sistema e aplicativo Diário de Bordo desenvolvido pela K13 Agência Web.

Imagem

Arraste e solte os arquivos aqui...
(ou clique para selecionar o(s) arquivo(s))

Selecionar arquivo... Procurar...

Voltar Atualizar

Figura 25: Página de Edição de Projeto

Fonte: Autoria própria

4.5.3 ÁREA DO ADMINISTRADOR

Quando um administrador entrar no sistema é direcionado para a página administrativa, isso é visível na Figura 26. E os *sidebars* da esquerda e o da direita são adaptados para ele, o da esquerda possui acesso para a área administrativa, lista de usuários e projetos cadastrados, enquanto que o da direita possui acessos para editar os dados da conta e sair do sistema.

Na página administrativa são apresentados gráficos e relatórios de usuários e projetos.

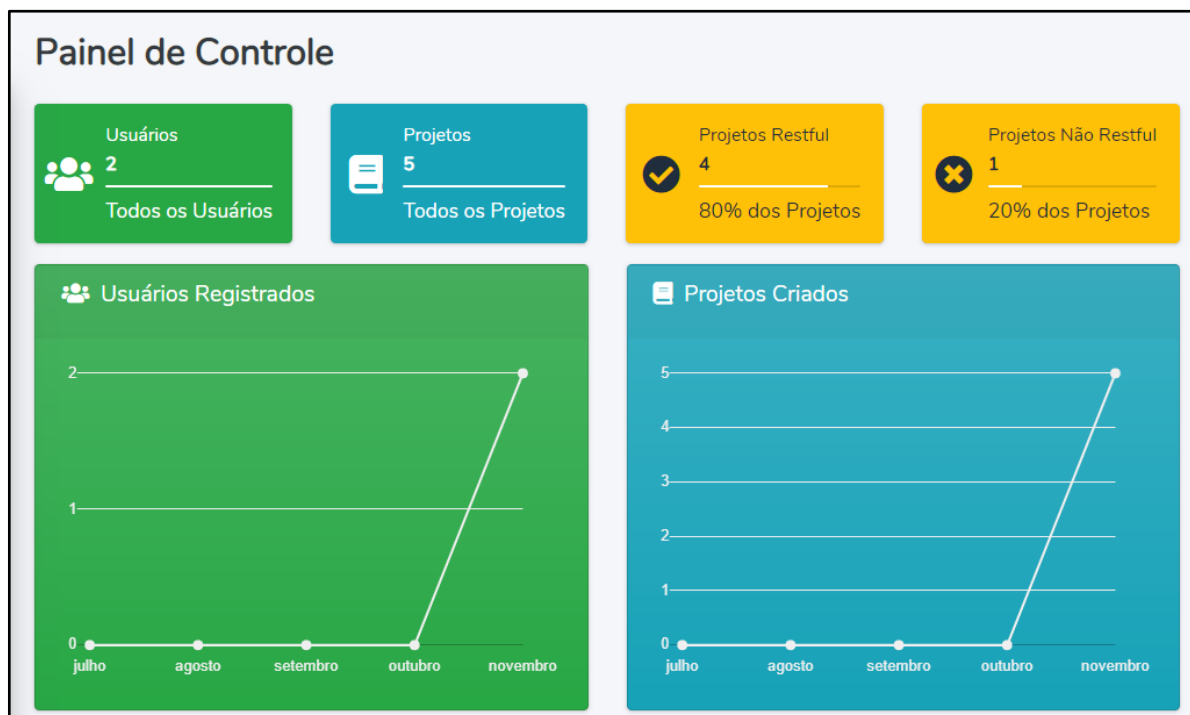


Figura 26: Página Administrativa

Fonte: Autoria própria

Nas Figuras 27 e 28 são apresentadas respectivamente as páginas de listagem e edição de contas de usuários.

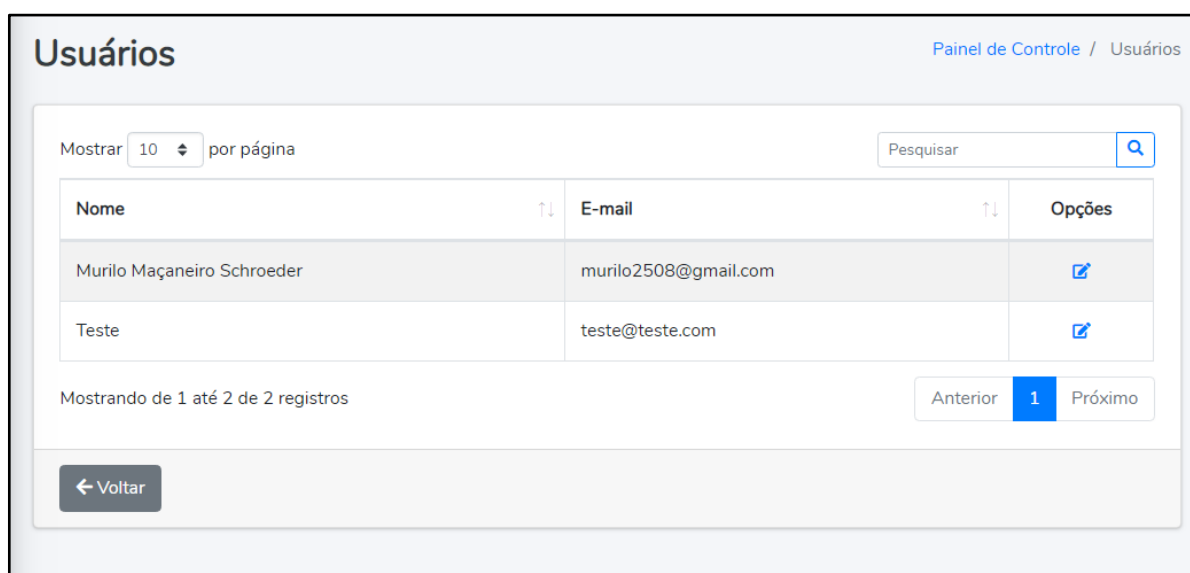


Figura 27: Página de Listagem de Usuários

Fonte: Autoria própria

A imagem mostra a interface de usuário para a edição de um perfil. No topo, há uma barra de navegação com o título 'Usuários' à esquerda e o caminho de navegação 'Painel de Controle / Usuários / Editar Usuário' à direita. Abaixo, um cabeçalho de seção indica 'Editar Usuário'. O formulário principal contém quatro campos de entrada: 'Nome' com o valor 'Teste', 'E-mail' com o valor 'teste@teste.com', 'Senha' (vazio) e 'Confirmação da senha' (vazio). Na base da interface, há dois botões: '← Voltar' em um botão cinza à esquerda e 'Atualizar' em um botão azul à direita.

Figura 28: Página de Edição de Usuário

Fonte: Autoria própria

Na Figura 29 é possível observar a listagem de projetos com os nomes, tipos e os usuários que criaram os projetos. E na Figura 30 é apresentada a página de edição de dados do administrador com os campos nome, e-mail e senha, similar às páginas de edição de usuários.

Projetos

Painel de Controle / Projetos

Mostrar 15 por página

Pesquisar

Nome	Tipo	Usuário	Opções
Integração Diário de Bordo	RESTful	Murilo Maçaneiro Schroeder	
Exemplo DocumentationJs	Não RESTful	Murilo Maçaneiro Schroeder	
Integração Mercado Livre K13	RESTful	Murilo Maçaneiro Schroeder	
Integração Loja Guarafer	RESTful	Murilo Maçaneiro Schroeder	
Integração Google Shopping	RESTful	Murilo Maçaneiro Schroeder	

Mostrando de 1 até 5 de 5 registros

Anterior1Próximo

Voltar

Figura 29: Página de Listagem de Projetos

Fonte: Autoria própria

Minha Conta

Painel de Controle / Minha Conta

Editar Minha Conta

Nome

Admin

E-mail

admin@admin.com

Senha

Confirmação da senha

Voltar

Atualizar

Figura 30: Página da Conta do Administrador

Fonte: Autoria própria

5 CONSIDERAÇÕES FINAIS

Neste trabalho foram levantados estudos das ferramentas similares ao sistema desenvolvido com o objetivo de resolver o problema de dificuldade na busca de informações sobre determinada linguagem de programação. Ao final, o sistema desenvolvido foi capaz de realizar a unificação de informações de documentações APIs que estavam dissipadas pela internet.

Neste TCC, a principal dificuldade foi o desenvolvimento e integração das ferramentas de geração da documentação API com o sistema. A principal importância atribuída a este trabalho estava na reunião de informações sobre diversas linguagens de programação em um único sistema, proporcionando um esclarecimento de dúvidas rápido e fácil.

5.1 TRABALHOS FUTUROS

Trabalhos futuros poderão ser realizados, como por exemplo, a adição de novas ferramentas que possuem a geração de arquivos Json e estrutura que sejam compatíveis com o sistema desenvolvido e seu banco de dados, o controle total sobre os projetos como edição e exclusão de projetos, classes, funções e comentários cadastrados e também a implementação de uma pesquisa *full text* para melhorar os resultados desejados. Além de plugins para editores de texto para a pesquisa de informações disponibilizadas pelo sistema.

REFERÊNCIAS

JONES, Darren. **JavaScript: Novice to Ninja** Second Edition. SitePoint, 2017. ISBN: 9780995382626

LARAVEL. **Introduction**. Disponível em <<https://laravel.com/docs/4.2/introduction>>. Acesso em: 16 out. 2019.

MAALEJ, Waleed; ROBILLARD, Martin P. “**Patterns of Knowledge in API Reference Documentation**”. IEEE TRANSACTIONS ON SOFTWARE ENGINEERING, v. 39, mar. 2013. Disponível em: <<http://ieeexplore.ieee.org/document/6473801/>>. Acesso em: 10 jun. 2018.

MDN WEB DOCS. **HTML**. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/HTML>>. Acesso em: 10 jun. 2018.

MDN WEB DOCS. **CSS**. Disponível em <<https://developer.mozilla.org/pt-BR/docs/Web/CSS>>. Acesso em: 10 jun. 2018.

MDN WEB DOCS. **What is JavaScript?**. Disponível em <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/First_steps/What_is_JavaScript>. Acesso em: 10 jun. 2018.

MICROSOFT. **Structured Query Language (SQL)**. Disponível em <<https://docs.microsoft.com/en-us/sql/odbc/reference/structured-query-language-sql?view=sql-server-2017>>. Acesso em: 11 jun. 2018.

OPPEL, Andy. **SQL: A Beginner's Guide**, Fourth Edition. McGraw Hill Professional, 2015. ISBN: 0071842578, 9780071842570

ORACLE. **How This API Document Is Organized**. Disponível em <<https://docs.oracle.com/javase/7/docs/api/>>. Acesso em: 10 jun. 2018.

PHP MANUAL. **PHP Manual**. Disponível em <http://php.net/manual/pt_BR/index.php>. Acesso em: 10 jun. 2018.

PHP. **O que é o PHP?**. Disponível em <https://www.php.net/manual/pt_BR/intro-what-is.php>. Acesso em: 16 out. 2019.

RUBY ON RAILS. **Ruby on Rails API**. Disponível em <<http://api.rubyonrails.org/>>. Acesso em: 10 jun. 2018.

W3. **Relationship to the World Wide Web and REST Architectures**. Disponível em <<https://www.w3.org/TR/ws-arch/#relwwwrest>>. Acesso em: 14 jun 2018.

W3SCHOOLS. **About Us**. Disponível em <<https://www.w3schools.com/about/default.asp>>. Acesso em: 10 jun 2018.

W3SCHOOLS. **CSS Introduction**. Disponível em

<https://www.w3schools.com/css/css_intro.asp>. Acesso em: 10 jun 2018.