

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

Murilo Maçaneiro Schroeder

**SAMP: SISTEMA DE AMPARO AO PROGRAMADOR**

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

GUARAPUAVA  
1º Semestre de 2018

Murilo Maçaneiro Schroeder

## **SAMP: SISTEMA DE AMPARO AO PROGRAMADOR**

Projeto de 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 – Câmpus Guarapuava, como requisito parcial para obtenção do título de Tecnólogo em Sistemas para Internet.

Orientador: Prof. Me. Guilherme da Costa Silva

Co-orientador: Prof. Dr. Luciano Ogiboski

GUARAPUAVA  
1º Semestre de 2018

## RESUMO

SCHROEDER, Murilo M.. SAMP: Sistema de Amparo ao Programador. 36 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2018.

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 complicado 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:** Programação. Documentação API. Desenvolvimento Web. Ruby. Ruby on Rails.

## ABSTRACT

SCHROEDER, Murilo M.. SAMP: Sistema de Amparo ao Programador. 36 f. Trabalho de Conclusão de Curso – Curso Superior de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2018.

When the goal is to program, probably the use of some kind of framework is indispensable and when used this presents a great amount 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 amount 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 documentation 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:** Programming. API Documentation. Web Development. Ruby. Ruby on Rails.

## LISTA DE TABELAS

Tabela 1 – Comparativo entre sistemas analisados.....	15
Tabela 2 – Cronograma.....	32

## 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.....	22
Figura 5 – Protótipo da página de cadastro de projetos.....	24
Figura 6 – Protótipo da página inicial.....	25
Figura 7 – Protótipo da página de projeto RESTful.....	26
Figura 8 – Protótipo da página de projeto não RESTful.....	27
Figura 9 – Protótipo da página de classe de projeto não RESTful.....	28
Figura 10 – Protótipo da página de funções de classe de projeto não RESTful.....	29
Figura 11 – Modelo do banco de dados.....	31

## 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

<b>1 INTRODUÇÃO</b>	<b>8</b>
1.1 OBJETIVOS	9
1.1.1 OBJETIVO GERAL	9
1.1.2 OBJETIVOS ESPECÍFICOS	9
<b>2 RESENHA LITERÁRIA</b>	<b>11</b>
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	16
2.2.3 CSS	16
2.2.4 RUBY	17
2.2.5 RUBY ON RAILS	17
2.2.6 SQL	17
2.2.7 RESTful	18
<b>3 METODOLOGIA</b>	<b>19</b>
3.1 ESTUDO DE TECNOLOGIAS	19
3.2 LEVANTAMENTO DE REQUISITOS	19
3.3 PROJETO DO SISTEMA	19
3.4 DESENVOLVIMENTO	20
3.5 TESTES	20
<b>4 DESENVOLVIMENTO PRELIMINAR</b>	<b>21</b>
4.1 REQUISITOS	21
4.1.1 REQUISITOS FUNCIONAIS	21
4.2 DIAGRAMAS DE CASOS DE USO	22
4.3 PROTÓTIPOS DE TELAS	22
4.4 MODELO DE BANCO DE DADOS	30
<b>5 CRONOGRAMA</b>	<b>32</b>
<b>6 CONSIDERAÇÕES FINAIS</b>	<b>33</b>
<b>REFERÊNCIAS</b>	<b>34</b>



## 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 então 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 será desenvolvido o Sistema de Amparo ao Programador (SAMP), onde será possível a visualização de documentações disponibilizadas pelos usuários do sistema. Possibilitando a busca por documentações de diversas linguagens de programação em único sistema.

## 1.1 OBJETIVOS

### 1.1.1 OBJETIVO GERAL

Desenvolver um sistema web de ajuda a programadores, com disponibilização de documentação APIs de diversas linguagens de programação em um único sistema fornecidas tanto pelo sistema quanto pelos próprios usuários. Além disso, facilitar a comunicação para que os usuários possam ajudar uns aos outros em relação a dúvidas e problemas.

### 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;
- Implementar plugin para interação do editor de textos Atom com o sistema proposto.

## **2 RESENHA LITERÁRIA**

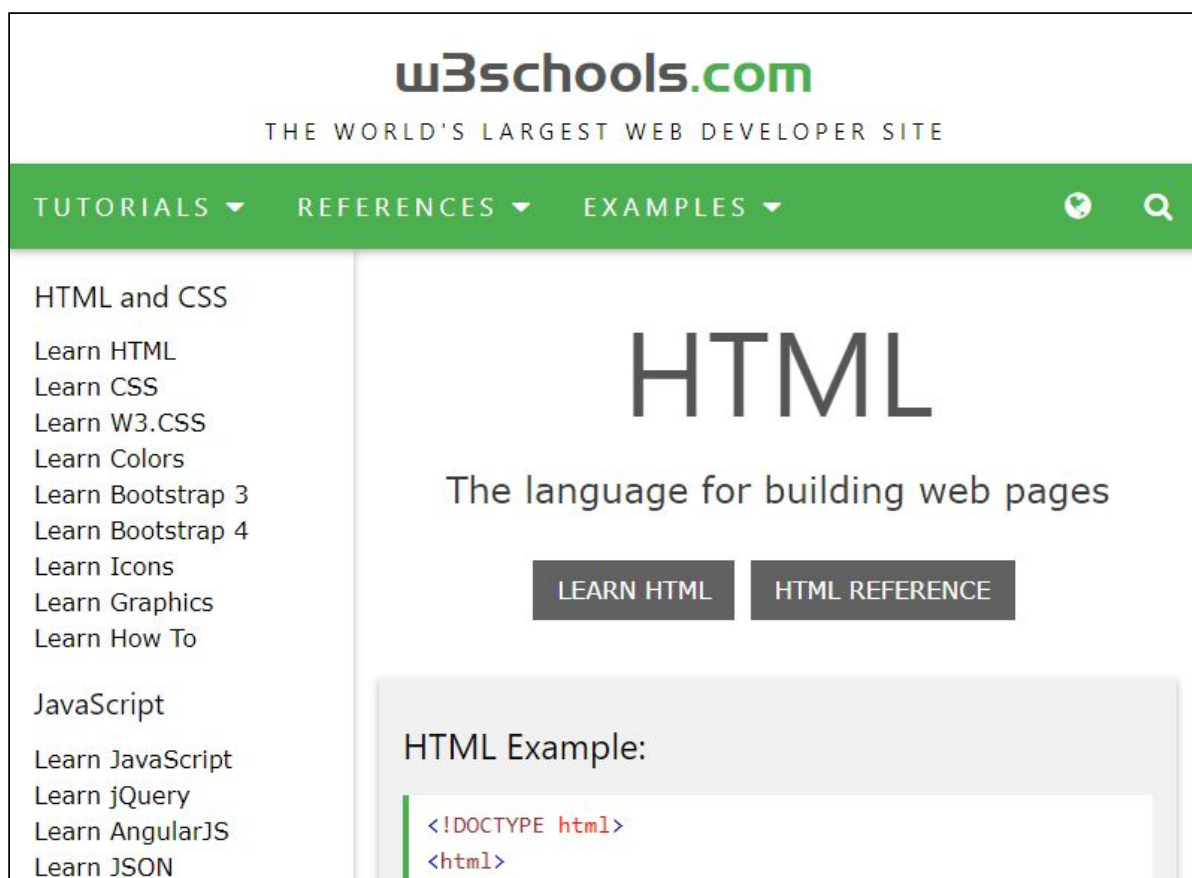
### **2.1 TRABALHOS RELACIONADOS**

O SAMP atenderá um grupo de usuários, provendo serviços de geração e disponibilização de documentações APIs. A seguir serão 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 proposto 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).

The image shows a screenshot of the Ruby on Rails API documentation. On the left is a navigation sidebar with a search bar and a list of classes. The 'Array < Object' class is selected and highlighted in red. The main content area on the right has a red header with the text 'Ruby on Rails 4.2.10' and 'Class Array < Object'. Below the header, a list of files is shown, including 'access.rb', 'conversions.rb', 'extract\_options.rb', 'grouping.rb', 'prepend\_and\_append.rb', 'wrap.rb', 'enumerable.rb', 'blank.rb', 'deep\_dup.rb', 'json.rb', and 'to\_query.rb'. Below this, a 'Methods' section is visible, listing methods such as 'deep\_dup', 'extract\_options!', 'fifth, forty\_two, fourth, from', and '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 seu exemplos com o intuito de ajudar e/ou facilitar a programação assim como proposto SAMP. (PHP, 2018).

The screenshot shows the PHP Manual page for the `strpos` function. The page is part of the 'String Functions' section. It includes a navigation bar with 'php', 'Downloads', 'Documentation', 'Get Involved', and 'Help' links, along with a search box. The breadcrumb trail is 'PHP Manual > Function Reference > Text Processing > Strings > String Functions'. A language selector is set to 'English'. The main content area features the function name `strpos` in a large font, followed by its availability in PHP 4, 5, and 7. The description states: 'strpos — Find the position of the first occurrence of a substring in a string'. A 'Description' section contains a code block showing the function signature: `int strpos ( string $haystack , mixed $needle [, int $offset = 0 ] )`. Below the code, it explains: 'Find the numeric position of the first occurrence of **needle** in the **haystack** string.' On the right side, there is a sidebar titled 'String Functions' listing various functions: `addslashes`, `addslashes`, `bin2hex`, `chop`, `chr`, `chunk_split`, `convert_cyr_string`, `convert_uuencode`, `convert_uuencode`, `count_chars`, `crc32`, `crypt`, `echo`, `explode`, `fprintf`, `get_html_`, and `translation_table`.

**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 proposto, 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, phpDocumentor, entre outros.

Tabela 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

## 2.2 FUNDAMENTAÇÃO TEÓRICA

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

### 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 RUBY

Ruby é uma linguagem dinâmica, onde segundo seu criador Yukihiro “Matz” Matsumoto foi baseada em Perl, Smalltalk, Eiffel, Ada e Lisp. Matz criou Ruby com o objetivo de ser mais robusto que Perl e mais orientada a objetos do que o Python. Ruby é flexível permitindo que seja alterada por quem está utilizando-a. É possível remover, alterar e adicionar novas funcionalidades a um objeto. (RUBY, 2018).

## 2.2.5 RUBY ON RAILS

Ruby on Rails é um *framework* que estende a linguagem de programação Ruby. Rails serve para a criação de websites através da combinação do Ruby, 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. (KEHOE, 2018).

## 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 proposto. Os passos metodológicos são descritos a seguir:

#### 3.1 ESTUDO DE TECNOLOGIAS

Nesta etapa serão estudados a linguagem Ruby e o *framework* Ruby on Rails que serão utilizados no desenvolvimento deste projeto, assim como também, o estudo de ferramentas de geração de documentação API. Será estudado a estrutura de arquivos, normalmente do tipo Json, gerados pelas ferramentas para que então as informações sejam interpretadas pelo sistema proposto.

#### 3.2 LEVANTAMENTO DE REQUISITOS

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

#### 3.3 PROJETO DO SISTEMA

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

para criação de protótipos.

### 3.4 DESENVOLVIMENTO

Durante esta etapa será realizada a implementação do sistema em forma de código de acordo com os resultados obtidos anteriormente, além de um plugin para o editor de texto Atom para pesquisa de informações disponíveis no sistema. Para o desenvolvimento será utilizado como base a linguagem Ruby juntamente com o *framework* Ruby on Rails e outros possíveis *frameworks*.

### 3.5 TESTES

Após o desenvolvimento é feito os testes e verificações das funcionalidades implementadas para que estejam de acordo com o que foi definido durante a etapa de levantamento de requisitos. E caso seja necessário uma refatoração do código será feita. Essa verificação será realizada através de testes unitários e de recursos.

## 4 DESENVOLVIMENTO PRELIMINAR

### 4.1 REQUISITOS

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

#### 4.1.1 REQUISITOS FUNCIONAIS

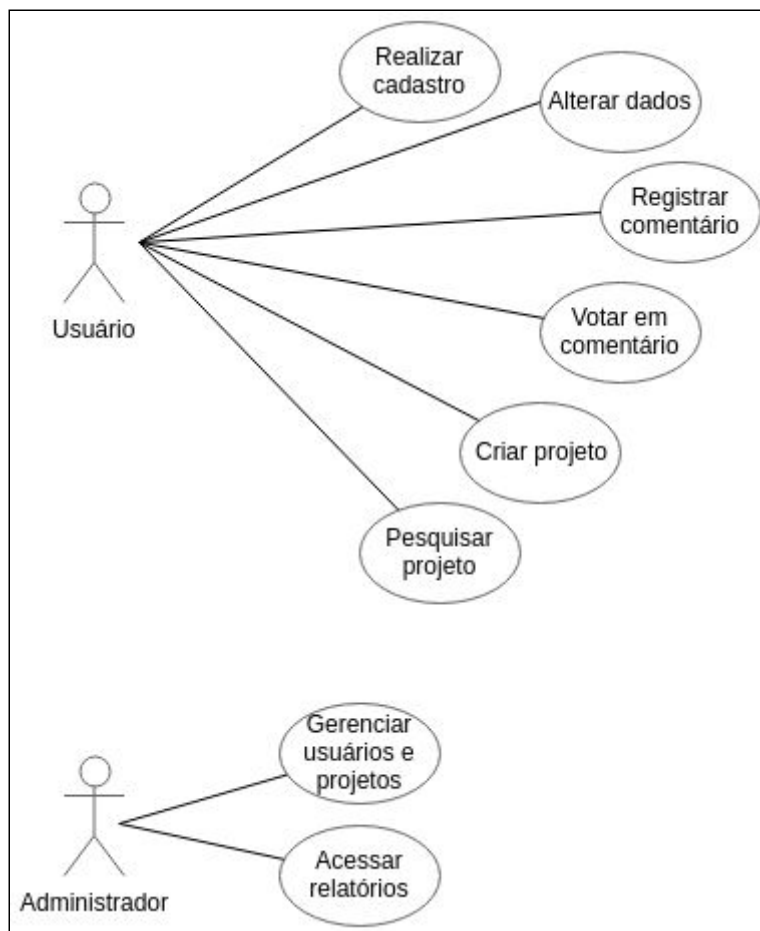
Nesta seção serão apresentados os requisitos funcionais:

- 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;

Será desenvolvido um plugin para o editor de texto Atom possibilitando o usuário ser redirecionado para uma página de pesquisa do sistema através de comandos do plugin.

## 4.2 DIAGRAMAS DE CASOS DE USO

A partir do 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 a serem desenvolvidas no sistema. Na Figura 5 é apresentada a página onde será 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 irá possuir 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 poderão postar seu exemplos de aplicação da função.



Criar Projeto

https://www.samp.com.br/users/projects/new

Logo

Pesquisar

Home Meus Projetos User 23 ▼

Home > Meus Projetos > Novo

**Criar Projeto**

Nome

Descrição

Linguagem

Ferramenta

Dados  Procurar

Imagem  Procurar

Meus Projetos

Perfil

Configurações

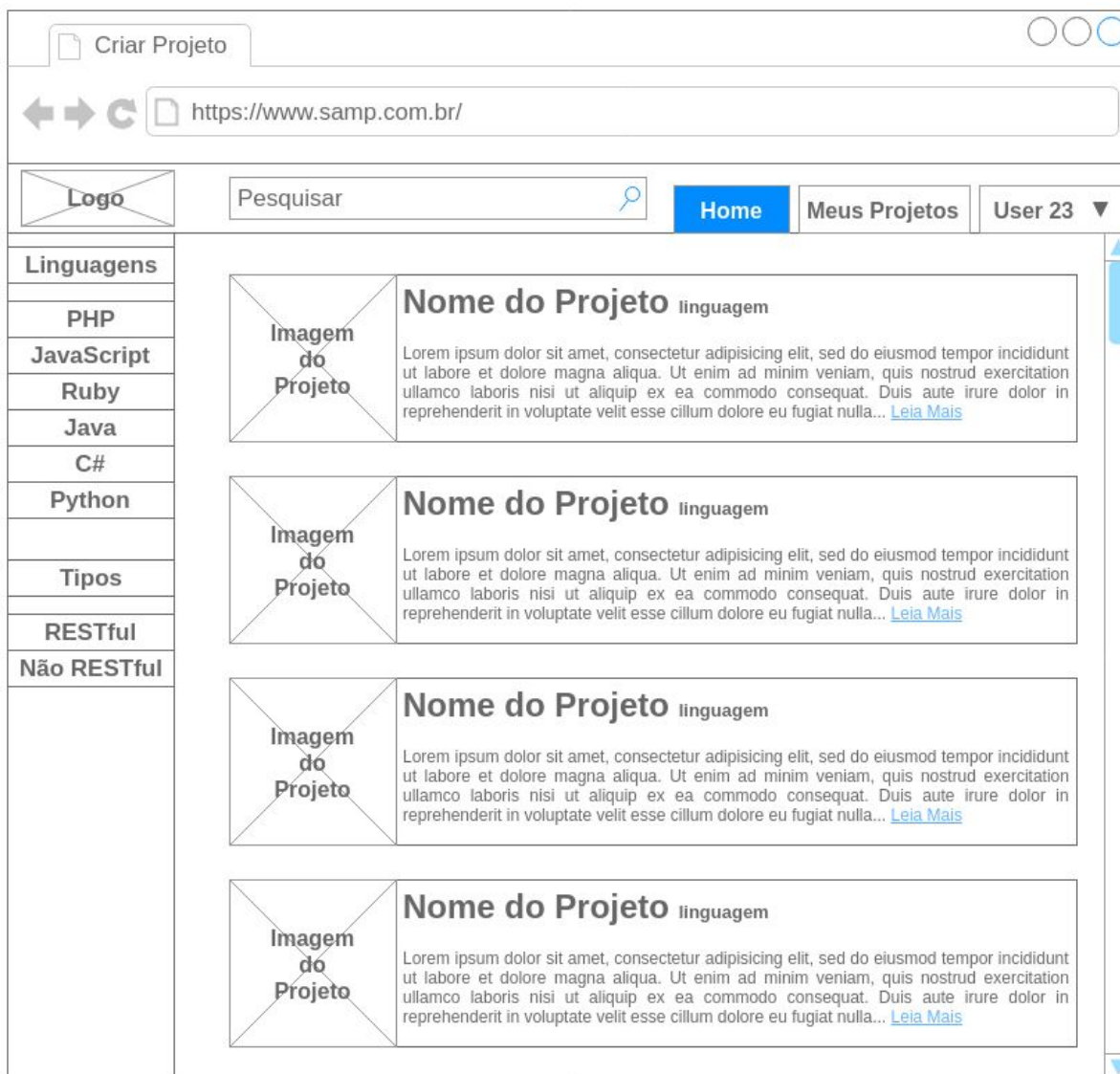
Perfil

Configuração

Sair

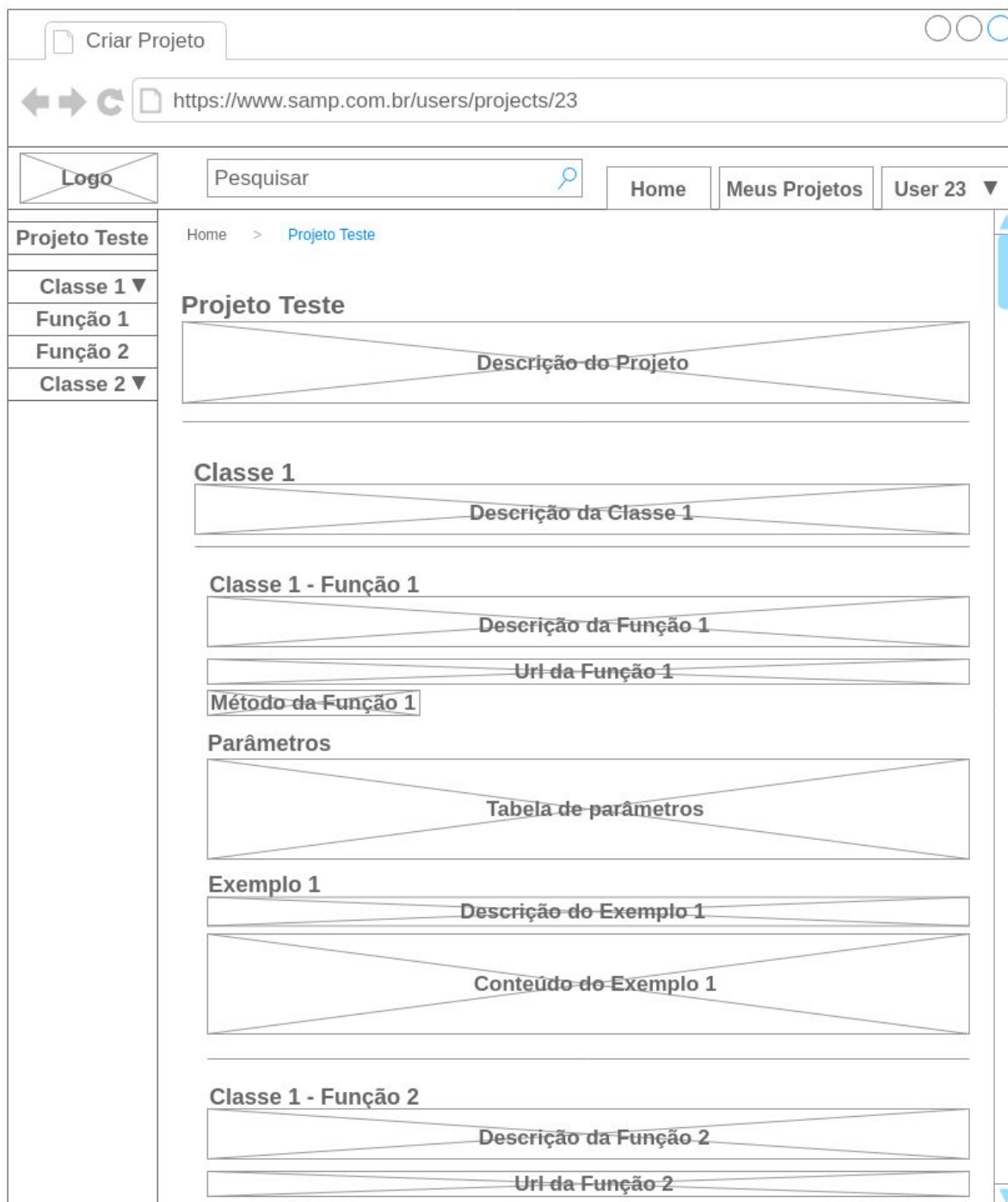
**Figura 5: Protótipo da página de cadastro de projetos**

Fonte: Autoria própria



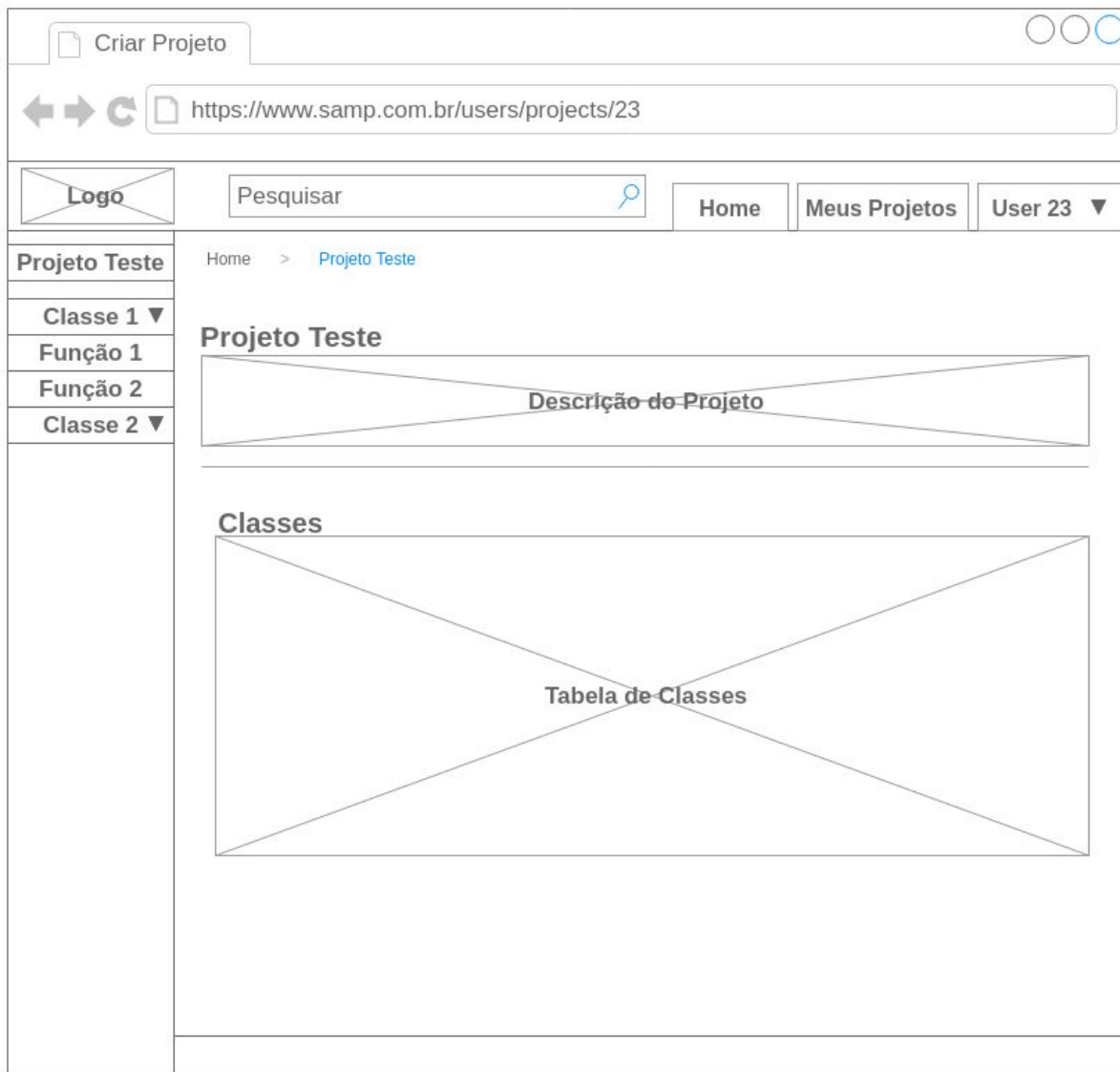
**Figura 6: Protótipo da página inicial**

Fonte: Autoria própria



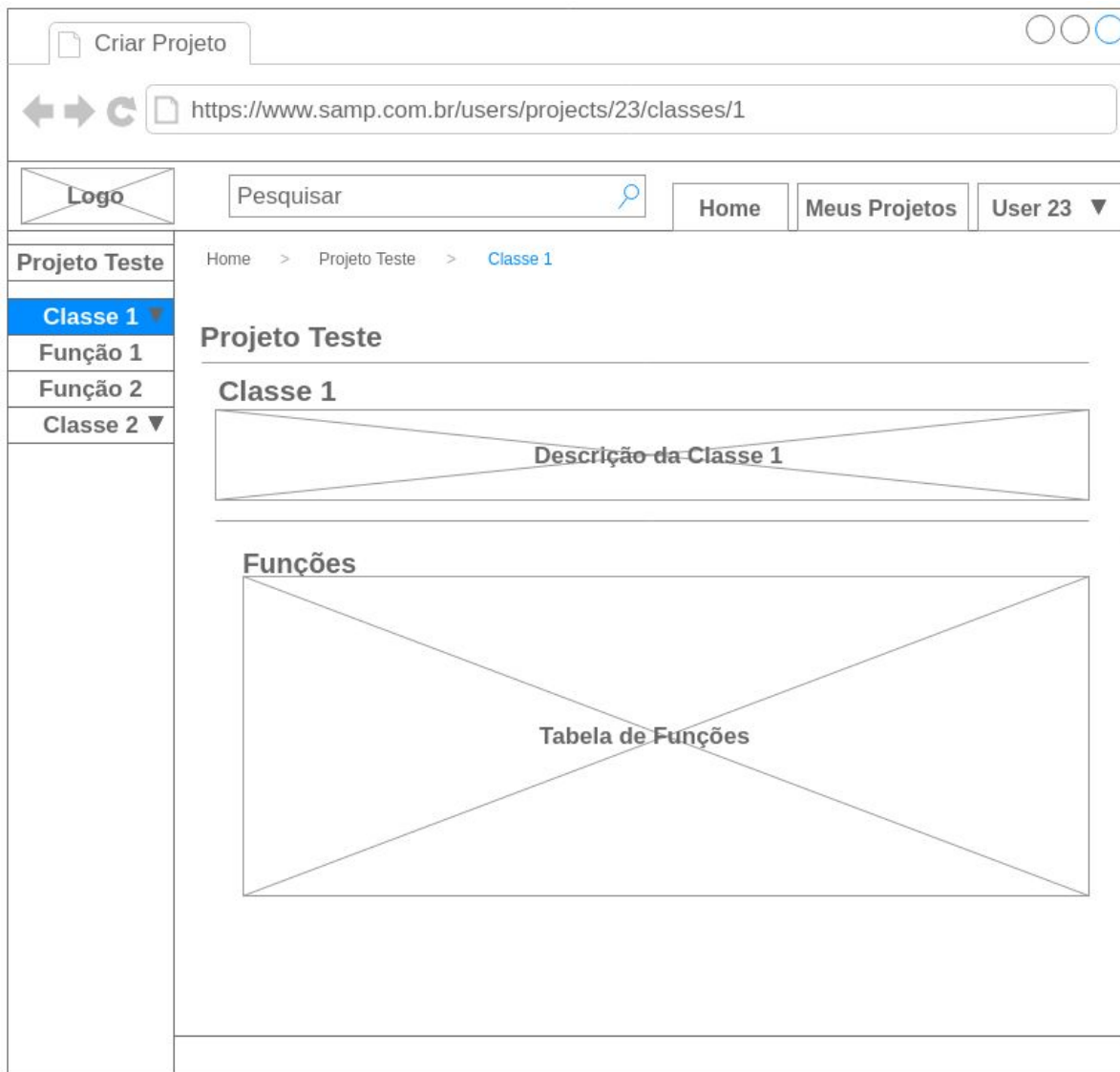
**Figura 7: Protótipo da página de projeto RESTful**

Fonte: Autoria própria



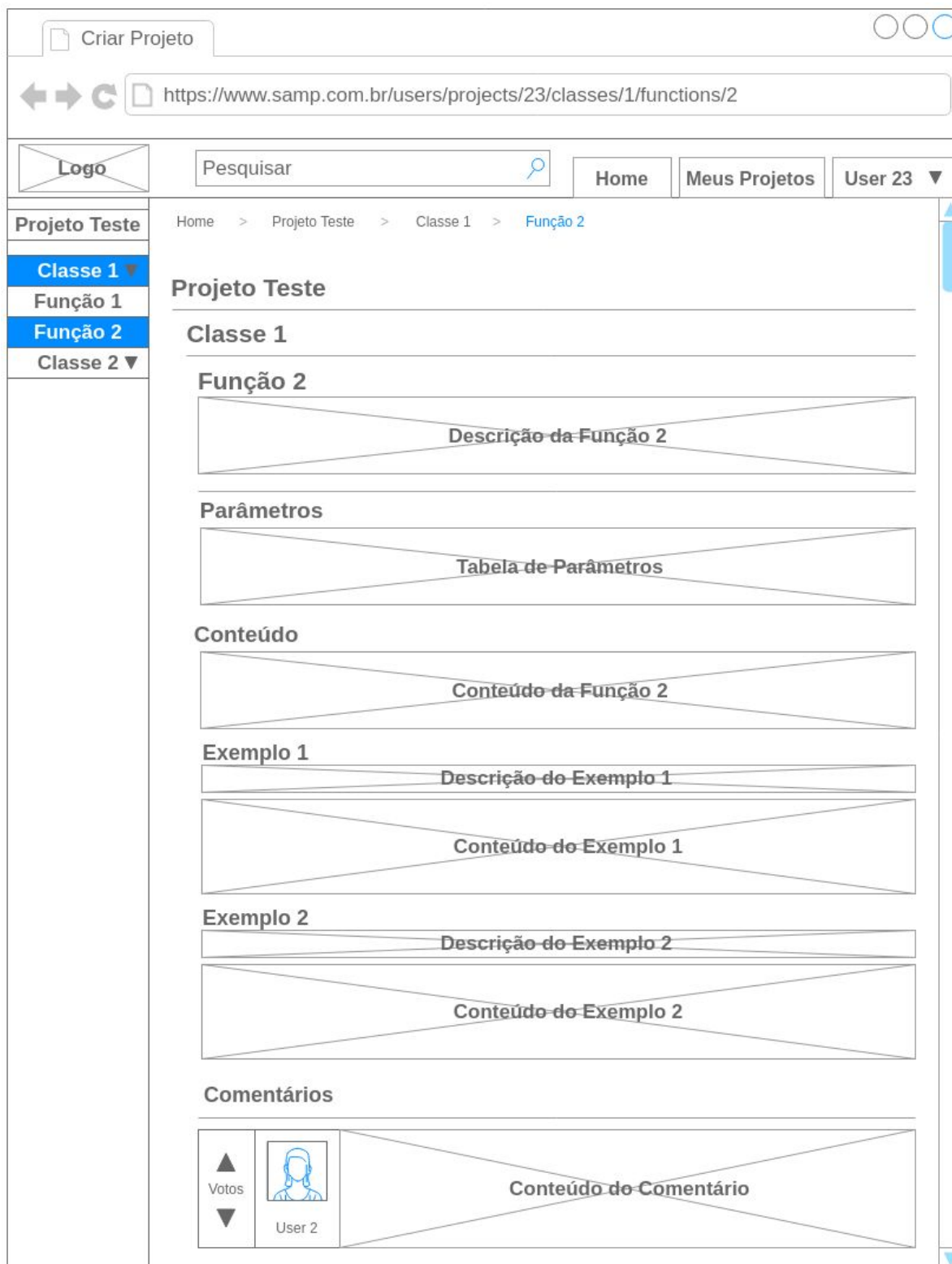
**Figura 8: Protótipo da página de projeto não RESTful**

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**

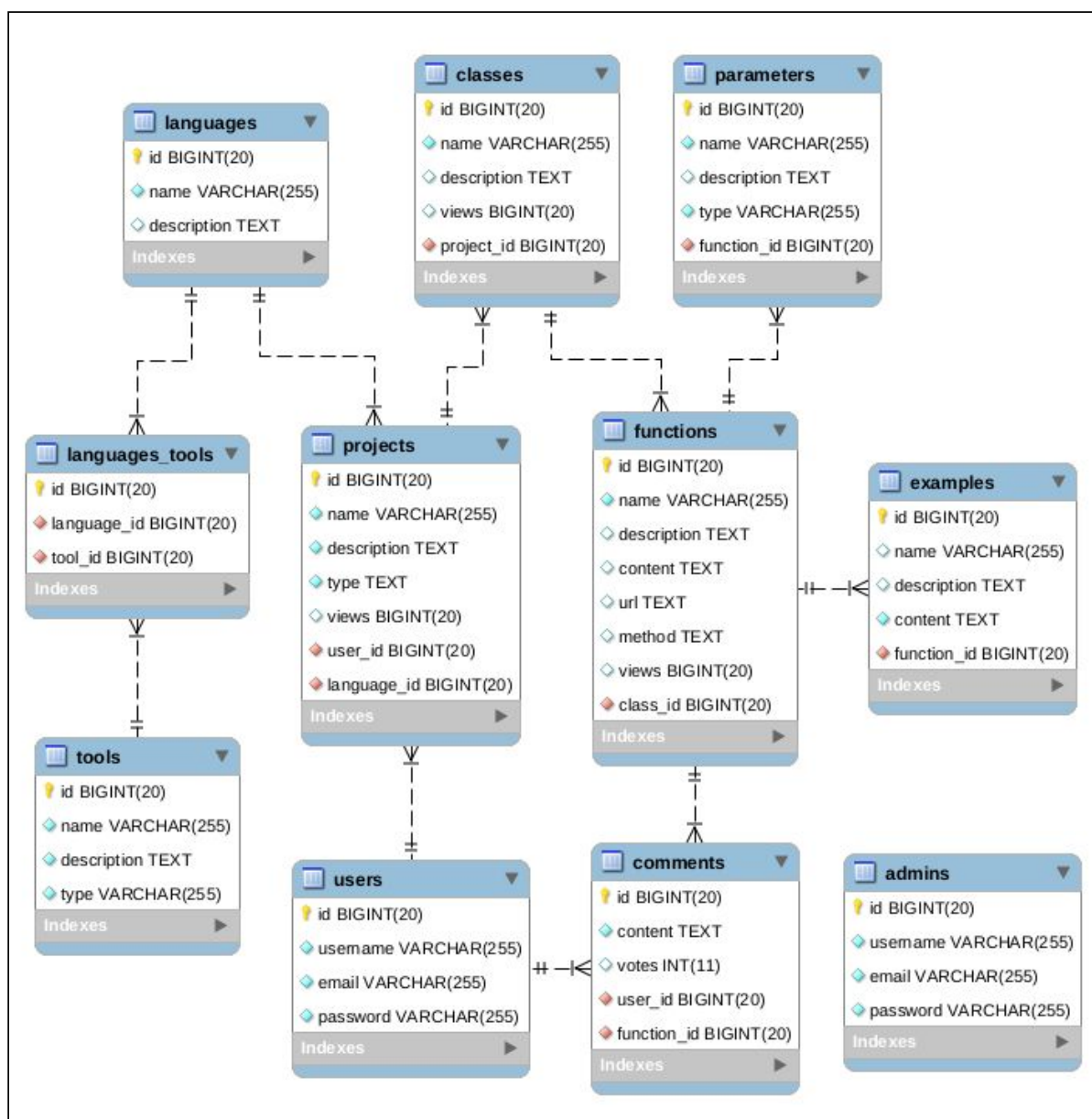
Fonte: Autoria própria

#### 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, phpDocumentor, documentationjs e ApiGen. Tendo como base a estrutura dos arquivo gerado pelas ferramentas foi feita a modelagem do banco de dados apresentado na Figura 11.

Informações sobre usuários comuns e administradores serão salvos respectivamente nas tabelas *users* e *admins*. Os usuários comuns poderão 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.

Note que durante o desenvolvimento do sistema algumas ferramentas podem ser adicionadas ou removidas e alguns atributos das tabelas do banco de dados podem ser alterados.



**Figura 11: Modelo do banco de dados**

Fonte: Autoria própria





## **6 CONSIDERAÇÕES FINAIS**

Neste trabalho foram levantados estudos das ferramentas similares ao sistema proposto com o objetivo de resolver o problema de dificuldade na busca de informações sobre determinada linguagem de programação. Neste projeto a principal dificuldade será o desenvolvimento e integração das ferramentas de geração da documentação API com o sistema. A principal importância atribuída a este trabalho está na reunião de informações sobre diversas linguagens de programação em um único sistema, proporcionando um esclarecimento de dúvidas rápido e fácil.

## REFERÊNCIAS

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

KEHOE, Daniel. **What is Ruby on Rails?**. Disponível em <<http://railsapps.github.io/what-is-ruby-rails.html>>. Acesso em: 10 jun. 2018.

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](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. **PHP Manual**. Disponível em <[http://php.net/manual/pt\\_BR/index.php](http://php.net/manual/pt_BR/index.php)>. Acesso em: 10 jun. 2018.

RUBY. **About Ruby**. Disponível em <<https://www.ruby-lang.org/en/about/>>. Acesso em: 10 jun. 2018.

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