

LARAHHELP: DESENVOLVIMENTO DE UM PROTÓTIPO DE GESTÃO DE BASE DE
CONHECIMENTO

LARAHHELP: DEVELOPMENT OF A PROTOTYPE OF KNOWLEDGE MANAGEMENT
BASE

Resumo. Uma base de conhecimento é uma biblioteca on-line de autoatendimento de informações sobre um produto, serviço, departamento ou tópico dentro de uma organização. Sendo a fundação para a prática de gestão do conhecimento. Visando o auxílio à organizações na prática de gestão do conhecimento, este trabalho propõe o desenvolvimento de uma abordagem tecnológica para gestão de uma base de conhecimento. Como resultado, o LaraHelp; um protótipo de gestão de uma base de conhecimento foi desenvolvido, o qual proporciona o acesso ao conhecimento da organização para auxiliar na tomada de decisões.

Palavras-chave: Laravel; Conhecimento; Base de Conhecimento; Gestão de Conhecimento.

Abstract. A knowledge base is an online self-service library of information about a product, service, department or topic within an organization. Being the foundation for knowledge management practice. Aiming to assist organizations in the practice of knowledge management, this work proposes the development of a technological approach for managing a knowledge base. As a result, LaraHelp; a knowledge base management prototype was developed, which provides access to the organization's knowledge to assist in decision making.

Keywords: Laravel; Knowledge; Knowledge Base; Knowledge Management.

INTRODUÇÃO

O sucesso nas organizações depende de boas decisões individuais e diárias que superam as decisões ruins ao longo do tempo. E o mais importante do que os gestores podem fazer para garantir o sucesso, é capacitar os funcionários em toda a organização a tomar boas decisões. Em parte, trata-se de simplesmente conceder autoridade para a tomada de decisões e estabelecer responsabilidade pelas decisões tomadas. É uma questão de prover as pessoas com o conhecimento necessário para tomarem boas decisões (NOVINS; ARMSTRONG, 1998).

Segundo ALAVI; LEIDNER, 1999), o conceito de codificar e transmitir conhecimento nas organizações não é novo onde programas de treinamento e desenvolvimento de funcionários, políticas organizacionais, rotinas, procedimentos, relatórios e manuais fazem essa função há muito tempo. Ao capturar, codificar e disseminar esse conhecimento, a organização reduz o nível de *know-how* necessário para seus gerentes e, ao mesmo tempo, melhora a eficácia e eficiência de suas operações.

A ênfase no conhecimento, nas organizações de hoje, se baseia no pressuposto de que barreiras à transferência e replicação do conhecimento lhe conferem importância estratégica. Muitas organizações estão desenvolvendo sistemas de informação projetados especificamente para facilitar o compartilhamento e a integração do conhecimento. Um sistema com essas características é denominado Sistema de Gestão do Conhecimento (*Knowledge Management System* - KMS) (ALAVI; LEIDNER, 1999).

Um componente integral de um KMS é chamado base de conhecimento. Uma base de conhecimento é usada para otimizar a coleta, organização e recuperação de informações para uma organização ou para o público em geral. Em relação à tecnologia da informação (TI), uma base de conhecimento é um recurso legível por máquina para a disseminação de informações, geralmente *on-line* ou com capacidade de ser *on-line* (ROUSE, 2020b).

Uma base de conhecimento bem organizada pode economizar o dinheiro de uma organização, diminuindo o tempo gasto pelos funcionários tentando encontrar informações. Como ferramenta de gestão de relacionamento com o cliente (*Customer Relationship Management* - CRM), uma base de conhecimento pode oferecer aos clientes

acesso fácil a informações que, de outra forma, exigiriam contato com a equipe de uma organização; via de regra, essa capacidade deve simplificar a interação tanto para o cliente quanto para a organização (ROUSE, 2020b).

Diante do exposto, surge a pergunta de pesquisa: Como facilitar a gestão de uma base de conhecimento de uma organização, sendo que muitas vezes más decisões não são tomadas pela inexistência do conhecimento e sim porque o conhecimento não estava disponível no tempo e lugar certo para ser utilizado?

Neste sentido, objetiva-se desenvolver um Protótipo para Gestão de uma Base de Conhecimento utilizando ferramentas de desenvolvimento *web*. Dessa forma espera-se facilitar a criação, utilização, análise e entendimento de uma base de conhecimento, proporcionando o acesso às informações que possam auxiliar na tomada de decisões.

Outros resultados buscados são: aprofundar os conhecimentos sobre KMS; conhecer e utilizar tecnologias e ferramentas *web*, tais como PHP, Laravel, jQuery, Bootstrap, entre outros e; aperfeiçoar o conhecimento acadêmico, por meio da leitura e compreensão de textos técnicos e científicos.

Este artigo, além desta seção introdutória apresenta outras seções, sendo elas: Fundamentação Teórica, Sistemas Similares, Materiais e Métodos, Resultados e Discussões, Considerações Finais e Trabalhos Futuros.

REVISÃO DE LITERATURA

Nesta seção é apresentada a fundamentação teórica, abordando os principais assuntos relacionados à esse trabalho: Conhecimento, Gestão do Conhecimento e KMS, Modelo Cliente-Servidor e Aplicações *Web*.

Conhecimento, Gestão do Conhecimento e KMS

Para definirmos KMS, é necessário primeiro definir: conhecimento e gestão do conhecimento. (PORFÍRIO, 2020) define conhecimento como o ato de entender, compreender, aprender algo por meio do raciocínio ou da experiência. O conhecimento está ligado à informação e o que pode se extrair a partir da mesma. Com a identificação de dados e entendendo seus significados é possível gerar o conhecimento, como uma informação que traz algum tipo de utilidade (SIGNIFICADOSBR, 2020).

(OLIVEIRA, 2020) apud [Carvalho 2012] afirma que informação é conhecimento, dado não é conhecimento e conhecimento não é dado definindo-os como:

- **Dado:** É o registro de um determinado evento de forma concreta ou abstrata.
- **Informação:** Conjunto de dados que apresenta um determinado significado dentro de um contexto.
- **Conhecimento:** É a informação tratada que muda um determinado comportamento, sendo resultado de um processamento complexo e subjetivo da informação.

Nesse contexto, a gestão do conhecimento é o nome de um conceito no qual uma empresa reúne, organiza, compartilha e analisa, de maneira consciente e abrangente, seus conhecimentos em termos de recursos, documentos e habilidades das pessoas. No início de 1998, acreditava-se que poucas empresas realmente tinham uma prática de gestão do conhecimento (ROUSE, 2020c).

Um plano de gestão de conhecimento envolve uma pesquisa dos objetivos organizacionais e uma análise das ferramentas, necessárias para atender os objetivos da organização. Onde o principal desafio é selecionar ou criar um KMS que se encaixe no contexto do plano geral e incentive os funcionários a usar o sistema e compartilhar informações (ROUSE, 2020c).

(CARVALHO; FERREIRA, 2013) discutem que o papel da TI na gestão do conhecimento consiste em aumentar a velocidade de transferência do conhecimento e também ampliar o seu alcance. Assim um KMS serviria para auxiliar o processo de estruturação e captura do conhecimento dos funcionários, disponibilizando-o em uma base comparilhada para toda a organização.

Com isso organizações começaram a implementar sistemas projetados especificamente para facilitar a codificação, coleta, integração e disseminação do conhecimento organizacional. Particularmente organizações baseadas em serviços e conhecimentos especializados, como por exemplo, empresas de consultoria de gestão e serviços profissionais. Sendo um sistema com essas características categorizado como um KMS (ALAVI; LEIDNER, 1999).

Em um KMS o maior desafio se encontra em como obter o conhecimento que se encontra disperso por toda a organização e conseguir representá-lo de uma forma acessível para todos (CARVALHO; FERREIRA, 2013). Assim o objetivo de um KMS é fornecer aos funcionários a capacidade de organizar e localizar conteúdo relevante e a experiência necessária para lidar com tarefas e projetos de negócios específicos (ROUSE, 2020b).

Modelo Cliente-Servidor

O modelo cliente-servidor, ilustrado na Figura 1, é uma estrutura que divide tarefas entre os provedores de um recurso ou serviço, chamados servidores, e solicitantes de serviço, chamados clientes. Esse modelo pode ser dividido em componentes, sendo eles (SOMMERVILLE, 2015):

- Um conjunto de servidores que ofereça um ou mais serviços a outros componentes.
- Um conjunto de clientes que utilizam os serviços oferecidos pelos servidores.
- Uma rede para permitir o acesso aos serviços pelos clientes.

Modelo Cliente-Servidor

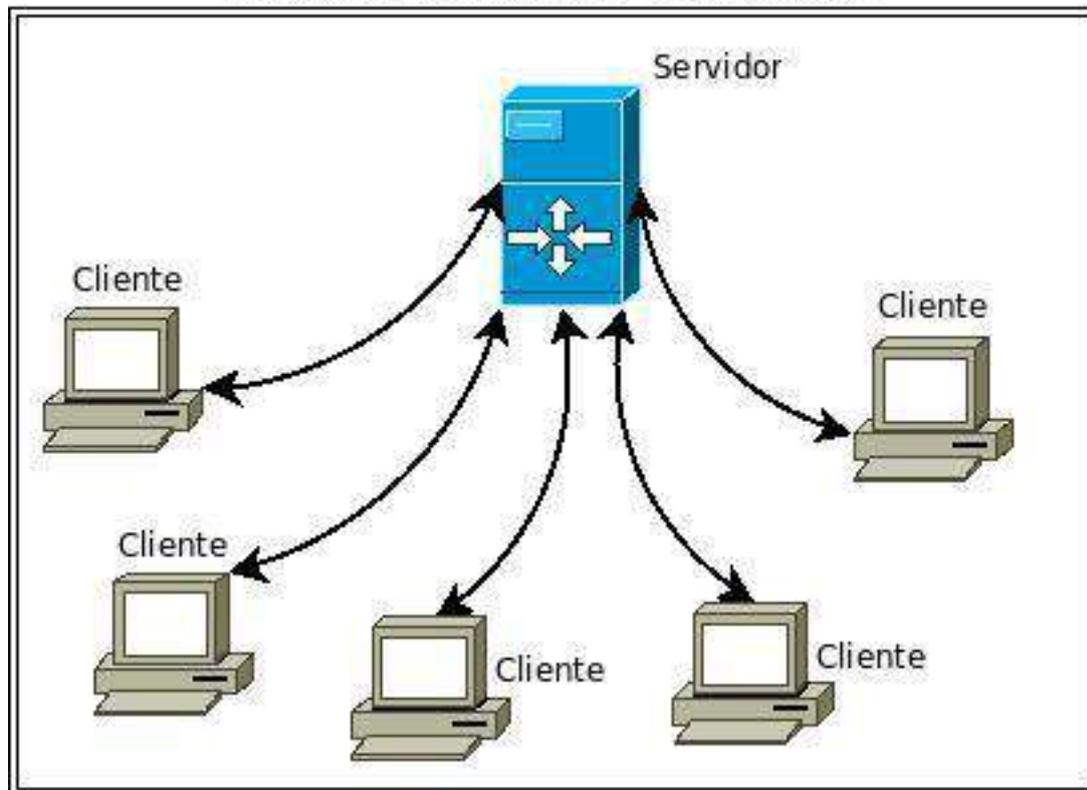


Figura 1. Modelo cliente-servidor (WIKILIVROS, 2020).

As requisições entre o servidor e os clientes são realizadas utilizando um protocolo, onde comumente o mais utilizado é o protocolo HTTP (*Hypertext Transfer Protocol*), que é a base de comunicação na Internet. Um grande benefício do modelo cliente-servidor é a separação e independência onde serviços e servidores podem ser modificados sem afetar outras partes (SOMMERVILLE, 2015).

Aplicações Web

Uma aplicação *web* é um programa armazenado em um servidor remoto e disponibilizado na Internet, normalmente por meio de um navegador *web*. Os serviços da Web são aplicativos da Web por definição e muitos, embora não todos, sites contêm aplicativos da Web. Qualquer componente de um site que execute alguma função para o usuário se qualifica como um aplicativo *web* (ROUSE, 2020a).

As aplicações *web* não precisam ser baixadas, pois são acessados na Internet. Para que uma aplicação *web* funcione, ela precisa de um servidor *web*, um servidor de aplicação, e se necessário um banco de dados. O servidor *web* gerencia as requisições dos clientes, enquanto o servidor de aplicativos realiza a tarefa solicitada. O banco de dados é utilizado para guardar as informações necessárias (ROUSE, 2020a).

Geralmente são divididas em camadas, cada uma com uma função distinta. As aplicações tradicionais geralmente são construídas em uma camada, que reside na máquina do cliente, mas as aplicações *web* geralmente são construídas em múltiplas camadas por natureza. Embora muitas variações sejam possíveis, a estrutura mais comum é a de três camadas sendo: Apresentação, aplicativo e armazenamento (PETERSEN, 2001).

As aplicações *web* existem desde antes mesmo da World Wide Web (WWW) ganhar popularidade. Onde, Larry Wall desenvolveu o Perl, uma linguagem de *script server-side* popular, em 1987. As primeiras aplicações *web* convencionais eram relativamente simples, mas no final dos anos 90 houve um impulso em direção a aplicações *web* mais complexas (NATIONS; MOUREAU, 2020).

2. Sistemas Similares

Alguns sistemas semelhantes para gestão de bases de conhecimento foram desenvolvidos. Dentre eles podemos destacar:

- **Opus¹**: Desenvolvido por Zeeshan Ahmad é um sistema de base de conhecimento de código aberto focado em equipes. Os funcionários podem criar, organizar e compartilhar conhecimento dentro de suas equipes e manter o trabalho organizado. Existe uma hierarquia estruturada e um poderoso mecanismo de pesquisa. Oferece também *templates* que ajudam na criação de documentos e geração de PDF (*Portable Document Format*) para facilitar o compartilhamento (AH-MAD, 2020).
- **Document360²**: Desenvolvido pela empresa Kovai.co é um sistema de base de conhecimento pago, focado nos funcionários e nos clientes da organização que utilizá-lo. Suas principais funcionalidades são (KOVAI.CO, 2020):
 - Versionamento dos artigos publicados.
 - Plataforma de discussão para os publicadores de artigos.
 - Configuração de SEO (Search Engine Optimization) para cada artigo.
 - Gerenciador completo de categorias.
 - Poderoso editor para escrita dos artigos.
 - Nível alto de segurança.

MATERIAIS E MÉTODOS

Nesta seção são apresentadas as metodologias e as ferramentas que foram utilizadas neste trabalho. As principais metodologias utilizadas neste trabalho são apresentadas a seguir.

Kanban

Adota-se a metodologia Kanban que sugere a utilização de um quadro visual ou físico, dividido em colunas que representam as etapas de um determinado processo. As tarefas a serem executadas no processo são divididas em cartões. Cada cartão contém a descrição da tarefa, a data de entrega e os responsáveis pela sua execução (GUEDES, 2020). Na Figura 2 é possível observar um exemplo de quadro Kanban.

¹Disponível em: <<https://github.com/ziishaned/opus>>

²Disponível em: <<https://document360.io/>>



Figura 2. Exemplo de quadro Kanban (ESPINHA, 2020)

O termo "Kanban" é originado de uma palavra japonesa que significa cartão visual. Inicialmente era usado para o controle de fluxo de peças e controle de estoque. Hoje, na área de desenvolvimento de software é usado para potencializar resultados (ESPINHA, 2020). É uma metodologia ágil e visual que tem como objetivo controle de produção ou gestão de tarefas. É ágil pois permite que as tarefas sejam controladas com rapidez, onde é possível acompanhar o ritmo de trabalho. É visual pois funciona por meio de um quadro com cartões e colunas como descrito anteriormente (ESPINHA, 2020).

Segundo (CURTO, 2020), as vantagens de utilização da metodologia Kanban são:

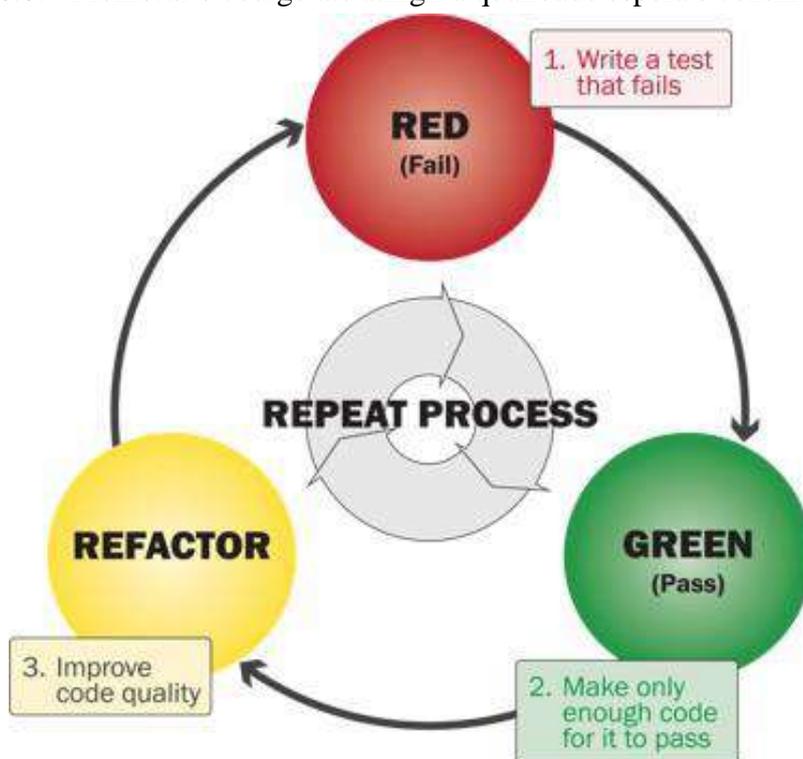
- Acompanhamento de tarefas em formato visual.
- Implementação de um fluxo de execução, permitindo aos responsáveis pela execução das tarefas, um maior controle sobre sua alocação.
- Acompanhamento de pendências de um usuário específico, em um portfólio de projetos.
- A integração com um cronograma permite um acompanhamento mais robusto de tarefas à medida que são movidas nas colunas.
- Gestão transparente que permite a equipe a determinar problemas no projeto.

Test-Driven Development (TDD)

Utiliza-se a técnica de desenvolvimento de software Test-Driven Development (TDD). O TDD relaciona os conceitos de verificação e validação e é baseado num ciclo curto de repetições (ANICHE, 2020). Segundo (BECK, 2003), o TDD incentiva pr objetos simples e inspira confiança na sua implementação. Com essa técnica código novo é somente im- plementado após a implementação um caso de teste automatizado que valide-o.

O ciclo de desenvolvimento usando TDD, ilustrado na Figura 3, pode ser abstraído em três etapas (BECK, 2003):

1. **Red:** Escrever um caso de teste automatizado que não funcione.
2. **Green:** Fazer o caso teste automatizado funcionar rapidamente, cometendo os pecados necessários no processo.
3. **Refactor:** Melhorar o código até atingir a qualidade esperada e continuar passando o



caso de teste.

Figura 3. Ciclo de desenvolvimento utilizando a técnica TDD (SILVA, 2020).

Assim é possível reduzir drasticamente a ocorrência de defeitos no código e tornar a funcionalidade testada clara para todos os envolvidos (BECK, 2003). Dentre os diversos ganhos utilizando a técnica TDD podemos citar (GAMA, 2020):

- *Feedback* rápido sobre a nova funcionalidade e as funcionalidades existentes.
- Códigos mais simples e limpos pois o código escrito é o mínimo para fazer o caso de teste automatizado passar.
- Segurança ao executar uma refatoração pois será possível verificar se a mesma afeta outras partes da aplicação.
- Produtividade elevada pois o desenvolvedor encontra menos erros e não precisa utilizar tempo de trabalho com depuradores de código.
- Código flexível pois com a escrita de casos de teste automatizados ele é dividido em partes menores para que sejam testáveis, assim o código é menos acoplado.

As principais ferramentas e tecnologias utilizadas neste trabalho são apresentadas a seguir.

Unified Modeling Language (UML)

A Unified Modeling Language (UML) é uma linguagem-padrão para a elaboração da estrutura de projetos de software. Ela pode ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software. Em outras palavras, na área de Engenharia de Software, a UML é uma linguagem de modelagem que permite representar um sistema de forma padronizada (com intuito de facilitar a compreensão e pré-implementação) (BOOCH; RUMBAUGH; JACOBSON, 2006).

A UML é adequada para a modelagem de sistemas, cuja abrangência pode incluir sistemas de informação corporativos a serem distribuídos a aplicações web e até sistemas embarcados complexos de tempo real. É uma linguagem muito expressiva, abrangendo todas as visões necessárias ao desenvolvimento e implantação desses sistemas (BOOCH; RUMBAUGH; JACOBSON, 2006).

Não é uma metodologia de desenvolvimento, portanto não mostra o que deve ser feito primeiro ou como deve ser projetado o sistema, mas auxilia a visualizar seu desenho e a comunicação entre os objetos. Basicamente, permite que desenvolvedores visualizem os produtos de seus trabalhos em diagramas padronizados (BOOCH; RUMBAUGH; JACOBSON, 2006).

Os objetivos da UML, segundo (BOOCH; RUMBAUGH; JACOBSON, 2006), são: especificação, documentação, estruturação para sub-visualização e maior visualização lógica do desenvolvimento completo de um sistema de informação.

PHPStorm

O PhpStorm³ é uma *Integrated Development Environment* (IDE) comercial para PHP que fornece um editor para PHP, HTML e JavaScript com análise de código em tempo real, prevenção de erros e automatizações, refatorações e testes para códigos PHP e JavaScript (JETBRAINS, 2020). A ferramenta, com a versão comercial na modalidade de estudante, foi

utilizada para o desenvolvimento do LaraHelp.

GIT

O GIT é um sistema de versionamento de código fonte distribuído, gratuito e de código aberto. Projetado para lidar com projetos grandes e pequenos com velocidade e eficiência. Foi desenvolvido inicialmente por Linus Torvalds para o desenvolvimento do kernel Linux (GIT, 2020).

Cada diretório de trabalho do GIT é um repositório com um histórico completo e habilidade total de acompanhamento das revisões, não dependente de acesso a uma rede ou a um servidor central. Existem diversos sites que disponibilizam hospedagem gratuita de código fonte para repositórios GIT e dentre eles está o Github⁴ que foi utilizado para hospedar o código fonte do LaraHelp.

³Disponível em: <<https://www.jetbrains.com/phpstorm/>>

⁴Disponível em: <<https://github.com/>>

PHP, HTML, CSS e JavaScript

Nesse trabalho, como se trata de uma aplicação web, foram utilizadas as linguagens de programação web PHP (acrônimo recursivo para PHP: *Hypertext Preprocessor*) e *JavaScript*. Para exibição e disposição dos elementos das páginas na web foram utilizadas as tecnologias *HyperText Markup Language* (HTML) e *Cascading Style Sheets* (CSS).

A linguagem PHP é uma linguagem de *script* de código aberto e de uso geral. É muito utilizada e adequada para o desenvolvimento web porque pode ser embutida dentro das páginas HTML. É uma linguagem fracamente tipada, ou seja, os tipos das variáveis não precisam ser declarados. Como o código PHP é mesclado nas páginas HTML ele é delimitado pelas *tags* `<?php e ?>` que marcam o início e fim respectivamente, de uma área de código em PHP (THE PHP GROUP, 2020).

Em complemento à linguagem PHP foi utilizada a linguagem *JavaScript*, a qual é uma linguagem de *scripts* orientada à objetos e multiplataforma. A diferença entre o PHP e o *JavaScript* é que o PHP somente é executado no servidor, enquanto o *JavaScript* pode ser executado no servidor com o uso de *frameworks* e no *host* (navegador) do usuário. O *JavaScript* permite um controle programático e dinâmico sobre os elementos de uma página HTML (MOZILLA DEVELOPER NETWORK, 2020).

MySQL

O MySQL⁵ é um Sistema Gerenciador de Banco de Dados (SGBD) de código aberto por meio da licença *General Public License* (GPL). Tem suporte de uma grande comunidade de entusiastas e desenvolvedores. Com comprovado desempenho, confiabilidade e facilidade de uso, o MySQL tornou-se a principal opção de banco de dados para aplicações web (ORACLE CORPORATION, 2020).

Principais funcionalidades do MySQL segundo (ORACLE CORPORATION, 2020):

- Portabilidade (suporta praticamente qualquer plataforma atual).
- Excelente desempenho e estabilidade.
- Pouco exigente quanto a recursos de hardware novos.

⁵Disponível em: <<https://www.mysql.com>>

- Facilidade no manuseio.
- Suporte a controle transacional.
- Suporte a *Triggers*.
- Suporte a *Stored Procedures* e *Functions*.
- Replicação facilmente configurável.
- Interfaces gráficas de fácil utilização.

Padrão Model-View-Controller (MVC)

O padrão MVC divide a aplicação em três partes fundamentais: Modelo (*Model*) que representa os dados da aplicação e as lógicas de negócios, Visão (*View*) que representa a informação recebida e enviada ao usuário e Controlador (*Controller*) que recebe as informações de entrada e controla o fluxo da aplicação (ALMEIDA, 2017). Na Figura 4 é possível observar o fluxo de eventos e informações no padrão MVC.

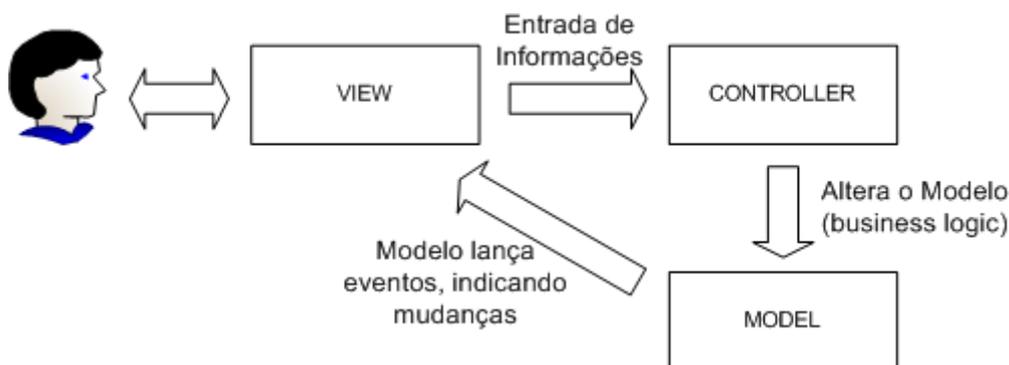


Figura 4. Fluxo de eventos e informações no padrão MVC (ALMEIDA, 2017).

Laravel

O Laravel⁶ é um *framework web* PHP gratuito e de código aberto, criado por Taylor Otwell e destinado ao desenvolvimento de aplicações web. O Laravel tenta facilitar as tarefas mais comuns no desenvolvimento da maioria dos sistemas web (LARAVEL LLC, 2020).

Dentre suas principais funcionalidades podemos destacar (CMARIX LTD., 2020):

- Mecanismo de *templates*, por meio do mecanismo de *templates* Blade.

- Mapeamento objeto-relacional (ORM - Object-relational mapping) por meio do Eloquent ORM.
- Suporte ao padrão MVC.
- Sistema de migração de banco de dados.
- Suporte facilitado para testes.
- Segurança.
- Quantidade de bibliotecas.
- Arquitetura modularizada.

jQuery

O jQuery⁷ é uma biblioteca de manipulação do Document Object Model (DOM) para JavaScript. O DOM é uma representação, em estrutura de árvore, de todos os elementos de uma página da web. O jQuery simplifica a sintaxe para encontrar, selecionar e manipular esses elementos do DOM (THE JQUERY FOUNDATION, 2020b).

Pode ser usado para encontrar um elemento no DOM com uma determinada propriedade (por exemplo, todos os elementos com uma *tag* p), alterar um ou mais de seus atributos (por exemplo, cor, visibilidade) ou fazer com que ele responda a um evento (por exemplo, um clique do mouse) (THE JQUERY FOUNDATION, 2020a).

Na Listagem 1 pode-se observar um exemplo de código utilizando jQuery. o código faz com que ao ser clicado no botão "Click", seja exibido um alerta com a frase "Button is clicked".

Listagem 1. Exemplo de código utilizando jQuery.

```
1 <button id="test">Click</button>2
3 <script>
4 $("#test").click(function() {
5     alert("Button is clicked");6 });
7 </script>
```

Bootstrap

O Bootstrap⁸ é um *framework* de HTML, CSS e JavaScript para desenvolvimento de projetos responsivos, sendo focado para dispositivos móveis na web. Torna o desenvolvimento *front-end* para *web* mais rápido e fácil (OTTO; THORNTON, 2020).

Docs UI Kit

O Docs UI Kit⁹ é um kit de interfaces de usuário para Bootstrap de código aberto por meio da licença MIT. O Kit disponibiliza várias páginas completas e funcionais e componentes reutilizáveis e personalizáveis. Cada componente é pensado para acelerar o fluxo de trabalho (HTMLSTREAM, 2020).

AdminLTE

O AdminLTE¹⁰ é um template de painel de controle, para administradores, construído com Bootstrap e fornece uma variedade de componentes responsivos, reutilizáveis e comumente usados. Suas principais vantagens são (ALMSAEED STUDIO, 2020):

- Design responsivo.
- Mais de 1000 ícones.
- 6 temas diferentes.
- Grande quantidade de *plugins*.
- Comunidade ativa.
- Compatibilidade com os navegadores mais populares.

DataTables

O DataTables¹¹ é um *plugin* do *JavaScript* que adiciona vários controles de interação avançados para tabelas nas páginas HTML. Suas principais funcionalidades são (SPRYMEDIA LTD., 2020):

⁸Disponível em: <<http://getbootstrap.com.br>>

⁹Disponível em: <<https://htmlstream.com/preview/docs-ui-kit/index.html>>

¹⁰Disponível em: <<https://adminlte.io>>

- Paginação, busca instantânea e ordenação por várias colunas.
- Suporta qualquer fonte de dados (DOM, *JavaScript*, AJAX e processamento no servidor).
- Grande variedade de temas disponíveis para as tabelas.
- Opções que podem ser extendidas e uma API simplificada.
- Código aberto por meio da licença MIT¹² e também suporte comercial.
- Testado com mais de 2900 testes de unidade.

Trumbowyg

O Trumbowyg¹³ é um editor de texto WYSIWYG (*What You See Is What You Get*) criado por Alexandre Demode. É distribuído como um *plugin* do jQuery e suas principais vantagens são (DEMODE, 2020):

- O editor e o código gerado são otimizados para suporte a HTML5 sendo compatível com todos os navegadores recentes.
- É rápido e leve, sem recursos inúteis, apenas contendo o mínimo de recursos necessários que um editor de texto precisa para gerar código limpo e semântico.
- Altamente customizável, onde praticamente todas as opções podem ser configuradas.
- Mais de 40 idiomas disponíveis para utilização.

4.2.1. PHPUnit

O PHPUnit¹⁴ é um *framework* de testes para PHP criado por Sebastian Bergmann. Se baseia na ideia de que os desenvolvedores devem poder encontrar erros no seu código rapidamente e também verificar que nenhuma regressão de código ocorreu em outras partes da base de código (BERGMANN, 2020). Na Listagem 2 é possível observar um exemplo de teste unitário utilizando o PHPUnit que valida a execução de uma operação de soma.

¹²Disponível em: <<https://opensource.org/licenses/MIT>> ¹³Disponível em: <<https://alex-d.github.io/Trumbowyg/>> ¹⁴Disponível em: <<https://phpunit.de/>>

Listagem 2. Exemplo de teste unitário no PHPUnit.

```
1 <?php2
3 class ExampleTest extends PHPUnit_Framework_TestCase {
4     public function testOnePlusOneEquals
5         Two ()
6     {
7         $result = 1 + 1;
8         $this->assertEquals(2, $result);
9     }
10
11 }
```

Heroku

Plataforma como serviço (*Platform as a Service* - PaaS) é uma categoria de serviços de computação em nuvem que fornece uma plataforma que permite aos clientes desenvolver, executar e gerenciar aplicações sem a complexidade de criar e manter a infraestrutura associada ao desenvolvimento e lançamento da mesma (BUTLER, 2013).

O Heroku¹⁵ é uma PaaS que suporta várias linguagens de programação. Está em desenvolvimento desde junho de 2007, quando suportava apenas a linguagem de programação Ruby. Atualmente suporta Java, Node.js, Scala, Clojure, Python, PHP e Go. As aplicações web executadas no Heroku geralmente têm um domínio exclusivo, normalmente "nome-da-aplicacao.herokuapp.com", usado como ponto de acesso para a mesma (SALESFORCE, 2020).

RESULTADOS E DISCUSSÕES

Nas próximas seções são apresentados e discutidos os resultados obtidos no desenvolvimento deste trabalho.

Modelagem do Protótipo

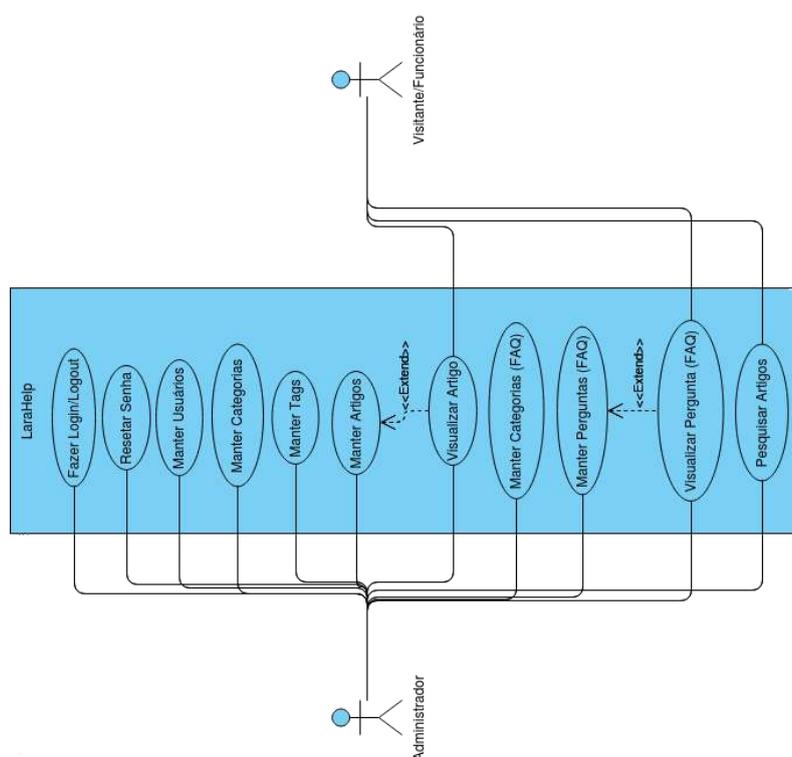
Dentre os inúmeros diagramas que podem ser desenvolvidos utilizando a UML, na modelagem do LaraHelp, foram utilizados os diagramas: Diagrama de Casos de Uso e Diagrama

¹⁵Disponível em: <<https://www.heroku.com>>

Entidade-Relacionamento (DER).

A modelagem de casos de uso está incorporada na UML para modelagem dos aspectos de interação de um sistema. Esta modelagem é de suma importância na descrição de pequenas interações que acontecem no sistema, estas interações são feitas por atores externos para com o sistema que está sendo desenvolvido (SOMMERVILLE, 2015).

Os atores podem ser usuários do sistema ou outros sistemas que estão integrados (SOMMERVILLE, 2015). A Figura 5 representa os casos de uso do LaraHelp. Os casos de uso “Manter” referem-se às ações de criar, excluir, editar e visualizar, por exemplo,



criar/excluir/editar/visualizar um artigo.

Figura 5. Diagrama de casos de uso do LaraHelp.

O DER é um diagrama de dados para descrever os dados ou aspectos de informação de um domínio de negócio ou seus requerimentos de processo, de uma maneira abstrata que em última análise pode ser implementada em um banco de dados (CHEN, 1976). Na Figura 11 no Apêndice B é possível observar o DER do LaraHelp.

Implementação do Protótipo

Como foi adotada a metodologia Kanban, cada funcionalidade ou correção a ser implementada no LaraHelp foi descrita em um cartão, também chamado de tarefa, colocado na coluna "A fazer". Quando inicia-se sua execução, o cartão é colocado na coluna "Em



andamento". Por fim, quando a tarefa é finalizada, o cartão é movido para a coluna "Concluído". Na Figura 6 é possível observar parte do quadro Kanban utilizado para gestão das tarefas do LaraHelp. O quadro foi criado e gerido por meio da ferramenta Trello¹⁶

Figura 6. Parte do quadro Kanban do LaraHelp.

Como descrito anteriormente na Seção 4.1.2, a implementação de cada tarefa do quadro foi realizada utilizando a técnica de TDD. Primeiramente foram implementados os casos de testes, que representam a tarefa em andamento. Após isso foi implementado o mínimo de código necessário para fazer com que os casos de teste passem. Por fim o código foi refatorado até atingir a qualidade desejada e os casos de teste continuarem passando.

Como dito anteriormente, o sistema de controle de versão GIT foi utilizado. Além de estar salvo localmente, o repositório foi salvo remotamente, no Github. Após a finalização de cada tarefa, ou parte de uma tarefa, foi realizado um *commit*. Na Figura 7 é possível observar parte dos *commits* que foram realizados e o comentário referente à cada um.

Cada *commit* salva e consolida o estado do projeto no GIT, ou seja, é como se fosse tirada uma foto de todos os arquivos naquele momento e armazenasse uma referência para essa captura. Para ser eficiente, se nenhum arquivo foi alterado, a informação não é armazenada novamente, apenas é criado um *link* para o arquivo idêntico anterior que já foi armazenado.

¹⁶Disponível em: <<https://trello.com>>

Implementado CRUD e testes de CRUD de categorias de perguntas frequentes Pr3d4dor committed 13 days ago	9d07826	↔
Adicionado seeder de categorias de perguntas frequentes Pr3d4dor committed 13 days ago	42b928b	↔
Implementado rotas CRUD de categorias de perguntas frequentes Pr3d4dor committed 13 days ago	98144b0	↔
Adicionado menu de perguntas frequentes e submenu categorias da mesma Pr3d4dor committed 13 days ago	c4a5d30	↔
Implementado CRUD e testes de CRUD de artigos Pr3d4dor committed 13 days ago	b884c6b	↔
Adicionado rotas CRUD de artigos Pr3d4dor committed 13 days ago	c2a78c5	↔

Figura 7. Parte dos *commits* realizados no repositório GIT do LaraHelp.

Na Figura 12 (Apêndice C) é apresentada a página principal do LaraHelp, onde é possível identificar as funcionalidades oferecidas para os visitantes/funcionários. O menu no canto superior direito disponibiliza as funcionalidades de: Ir para a página principal, visualizar artigos, visualizar categorias, visualizar *tags*, visualizar perguntas frequentes e acessar o painel de administrador. É possível também observar a funcionalidade de pesquisar artigos por meio da barra de pesquisa.

Um exemplo de pesquisa de artigos, é apresentado na Figura 13 (Apêndice C). A pesquisa pode ser feita em uma categoria específica de artigos, em artigos que contêm uma ou mais *tags* e por uma determinada frase de busca. Um artigo presente nos resultados da pesquisa pode ser visualizado, conforme a Figura 14 (Apêndice C), clicando no *link* "Ler Mais". A página de listagem de categorias de artigos pode ser observada na Figura 15 (Apêndice C) enquanto a página de listagem *tags* de artigos é ilustrada na Figura 16.

A página de perguntas frequentes (*Frequently Asked Questions* - FAQ) é apresentada na Figura 17 (Apêndice C). No menu à direita estão dispostas todas as categorias de perguntas frequentes que contêm pelo menos uma pergunta associada. Ao ser clicado em uma das categorias de perguntas, são exibidas todas as perguntas associadas à mesma.

O acesso ao painel administrativo é realizado pela página de *login*, representada pela Figura 18 (Apêndice C). Nessa página o administrador também tem acesso à funci-

onalidade de resetar senha, representada na Figura 19. Após realizado o *login* o administrador é redirecionado para a página de *dashboard* observada na Figura 20 (ApêndiceC).

Nas Figuras 21 à 24 do Apêndice C, é possível identificar respectivamente as funcionalidades de cadastrar, excluir, editar e visualizar um artigo. Essa mesma padronização foi seguida nas demais funcionalidades (Usuário, Categoria, entre outros).

O código fonte completo do LaraHelp é código aberto e está disponível em um repositório no GitHub¹⁷. Uma instância de exemplo do LaraHelp está disponível na Internet hospedado na plataforma Heroku¹⁸.

Suíte de Casos de Teste



Devido à utilização da técnica TDD, também como resultado foi obtido uma suíte de casos de teste. A suíte de casos de teste valida se as funcionalidades desenvolvidas estão funcionando corretamente. Na Figura 8 é possível observar toda a suíte de casos de testes implementada.

Figura 8. Suíte de casos de teste do LaraHelp.

Ainda na Figura 8 é possível observar que os casos de testes foram categorizados

¹⁷Disponível em: <<https://github.com/Pr3d4dor/larahelp>>

¹⁸Disponível em: <<https://larahelp.herokuapp.com>> - Credenciais para login no painel administrativo: email: admin@admin.com, senha: admin

em três tipos, sendo eles:

- **Funcionalidade (*Feature*):** Esses casos de teste validam se uma funcionalidade da aplicação, como por exemplo editar um artigo, está funcionando corretamente.
- **Navegador (*Browser*):** Esses casos de teste fazem a mesma coisa que os casos de teste de *feature* porém são executados por meio de um navegador *web*.
- **Unitário (*Unit*):** Esses casos de teste validam se uma parte individual do código, como por exemplo um método, está funcionando corretamente.

Exemplos de casos testes de funcionalidade, navegador e unitário, podem ser observados

```

gblne@owl:ot:~/Documents/GitHub/larahelp$ composer test
> vendor/bin/phpunit
PHPUnit 6.5.2 by Sebastian Bergmann and contributors.

.....
120 / 120 (100%)

Time: 1.56 minutes, Memory: 44.00 MB

OK (120 tests, 345 assertions)

Generating code coverage report in Clover XML format ... done [134 ms]
Generating code coverage report in HTML format ... done [457 ms]

```

respectivamente nas Listagens 3 à 5 (Apêndice A). A suíte completa de casos testes pode ser consultada no repositório do LaraHelp¹⁹, onde na Figura 9 é possível observar a sua execução e o resultado obtido.

Figura 9. Resultado da execução da suíte de casos testes do LaraHelp.

Ainda na Figura 9 é possível observar que a execução da suíte de testes gera um relatório de



cobertura de código. (ROCHA, 2020) define cobertura de código como uma métrica utilizada para medir a quantidade de linhas de código que foram validadas pela suíte de testes. Na Figura 10 é apresentado o relatório de cobertura de código.

Figura 10. Relatório de cobertura de código gerado pela suíte de testes do LaraHelp.

¹⁹Disponível em: <<https://github.com/Pr3d4dor/larahelp/tree/master/tests>>

CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho teve como objetivo principal desenvolver um protótipo para gestão de uma base de conhecimento, utilizando ferramentas de desenvolvimento *web*. Diante do ex- posto nas seções anteriores, chegou-se a uma versão funcional de um protótipo de uma aplicação *web*, que foi denominado LaraHelp, que cumpre os objetivos propostos.

O desenvolvimento do LaraHelp foi facilitado por meio de *frameworks* e bibliotecas disponíveis na *web*. O uso desses *frameworks* e bibliotecas trouxe economia de tempo. Onde caso não existissem, a conexão com o MySQL e a paginação de resultados por exemplo, teriam que ser implementados manualmente. Isso demonstra a importância do código aberto e do reúso.

O LaraHelp pode ainda ser estendido e aprimorado. Como futuras melhorias sugere-se:

1. Implementação de um sistema de traduções para permitir o uso em países além do Brasil.
2. Utilizar princípios da Interação Humano-Computador (IHC) para melhorar as interfaces do protótipo, facilitando o uso do sistema e agregando acessibilidade ao sistema, permitindo que pessoas com algum tipo de necessidade especial (física, motora, entre outras) possam utilizar o sistema.
3. Realizar testes de interface e melhorá-la com os resultados obtidos.
4. Desenvolvimento e escrita de um manual de usuário para facilitar a utilização.

REFERÊNCIAS

- _____. **What is knowledge base?** 2020. <<https://whatis.techtarget.com/definition/knowledge-base>>. [Online; acessado em 21 jan. 2020].
- _____. **What is knowledge management (KM)?** 2020. <<https://searchdomino.techtarget.com/definition/knowledge-management>>. [Online; acessado em 22 jan. 2020].
- _____. **jQuery.** 2020. <<https://jquery.com/>>. [Online; acessado em 20 jan. 2020]. <<http://getbootstrap.com.br/>>. [Online; acessado em 15 jan. 2020]. <<https://alex-d.github.io/Trumbowyg/>>. [Online; acessado em 22 jan. 2020].
- AHMAD, Z. **Opus - Opensource knowledge base application for Teams.** 2020. <<https://github.com/ziishaned/opus>>. [Online; acessado em 18 fev. 2020].
- ALAVI, M.; LEIDNER, D. Knowledge management systems: issues, challenges, and benefits. **Communications of the Association for Information systems**, v. 1, n. 1, p. 7, 1999.
- ALMEIDA, R. R. de. **Model-View-Controller (MVC).** 2017. <<http://www.dsc.ufcg.edu.br/~jacques/cursos/map/html/arqu/mvc/mvc.htm>>. [Online; acessado em 15 jan. 2017].
- ALMSAEED STUDIO. **AdminLTE Bootstrap Admin Dashboard Template.** 2020. <<https://adminlte.io/>>. [Online; acessado em 20 jan. 2020].
- ANICHE, M. **Test-Driven Development.** 2020. <<https://tdd.caelum.com.br/>>. [Online; acessado em 20 jan. 2020].
- BECK, K. **Test-driven development: by example.** [S.l.]: Addison-Wesley Professional, 2003.

BERGMANN, S. **PHPUnit - The PHP Testing Framework**. 2020. <<https://phpunit.de/>>. [Online; acessado em 20 jan. 2020].

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **UML: guia do usuário**. [S.l.]: Elsevier Brasil, 2006.

BUTLER, B. Paas primer: What is platform as a service and why does it matter. **Network World**, v. 11, 2013.

CARVALHO, R. B. d.; FERREIRA, M. A. T. **Tecnologia da informação aplicada à gestão do conhecimento**. 2013. <<http://repositorios.questoesemrede.uff.br/repositorios/bitstream/handle/123456789/1445/Tecnologia%20-%20Carvalho.pdf>>. [Online; acessado em 22 jan. 2020].

CHEN, P. P.-S. The entity-relationship model—toward a unified view of data. **ACM Transactions on Database Systems (TODS)**, Acm, v. 1, n. 1, p. 9–36, 1976.

CMARIX LTD. **9 Benefits of Laravel Application Development**. 2020. <<https://www.cmarix.com/9-benefits-of-laravel-application-development/>>. [Online; acessado em 20 jan. 2020].

CURTO, H. **Cinco benefícios do kanban para a gestão de projetos**. 2020. <<https://netproject.com.br/blog/cinco-beneficios-do-kanban-para-a-gestao-de-projetos/>>. [Online; acessado em 20 jan. 2020].

DEMODE, A. **Trumbowyg - A lightweight WYSIWYG editor by Alex-D**. 2020.

ESPINHA, R. G. **Kanban - O que é e TUDO sobre como gerenciar fluxos de trabalho**. 2020. <<https://www.treinaweb.com.br/blog/principios-basicos-do-kanban/>>. [Online; acessado em 20 jan. 2020].

GAMA, A. **Test Driven Development: TDD Simples e Prático**. 2020. <<https://www.devmedia.com.br/test-driven-development-tdd-simples-e-pratico/18533>>. [Online; acessado em 20 jan. 2020].

GIT. **Git - git Documentation**. 2020. <<https://git-scm.com/docs/git>>. [Online; acessado em 14 jan. 2020].

GUEDES, M. **Princípios básicos do Kanban**. 2020. <<https://www.treinaweb.com.br/blog/principios-basicos-do-kanban/>>. [Online; acessado em 20 jan. 2020].

HTMLSTREAM. **Documentation & Help Desk - A beautiful Open Source Bootstrap 4 UI Kit!** 2020. <<https://htmlstream.com/preview/docs-ui-kit/index.html>>. [Online; acessado em 20 jan. 2020].

JETBRAINS. **Features – PhpStorm**. 2020. <<https://www.jetbrains.com/phpstorm/features/>>. [Online; acessado em 14 jan. 2020].

KOVAICO. **Self Service Knowledge Base Software Features - Document360**. 2020. <<https://document360.io/features/>>. [Online; acessado em 18 fev. 2020].

LARAVEL LLC. **Laravel - The PHP Framework for Web Artisans**. 2020. <<https://laravel.com/>>. [Online; acessado em 20 jan. 2020].

MOZILLA DEVELOPER NETWORK. **Introdução - JavaScript - MDN**. 2020. <<https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Guide/Introduction>>. [Online; acessado em 18 jan. 2020].

- NATIONS, D.; MOUREAU, E. **What Exactly Is a Web Application?** 2020. <<https://www.lifewire.com/what-is-a-web-application-3486637>>. [Online; acessado em 21 jan. 2020].
- NOVINS, P.; ARMSTRONG, R. Choosing your spots for knowledge management. **Perspectives on Business Innovation**, v. 1, p. 45–54, 1998.
- OLIVEIRA, C. **Mas afinal o que é conhecimento?** 2020. <<https://imasters.com.br/devsecops/mas-afinal-o-que-e-conhecimento>>. [Online; acessado em 22 jan. 2020].
- ORACLE CORPORATION. **MySQL | O Banco de Dados de Código Aberto Mais Popular.** 2020. <<https://www.oracle.com/br/mysql/index.html>>. [Online; acessado em 19 jan. 2020].
- OTTO, M.; THORNTON, J. **Bootstrap Framework – O mais popular framework front-end responsivo e focado para dispositivos móveis do mundo.** 2020.
- PETERSEN, J. Benefits of using the n-tiered approach for web applications. **URL: <http://www.adobe.com/devnet/coldfusion/articles/ntier.html>**, 2001.
- PORFÍRIO, F. **Conhecimento: entenda melhor esse conceito.** 2020. <<https://mundoeducacao.bol.uol.com.br/filosofia/conhecimento.htm>>. [Online; acessado em 22 jan. 2020].
- ROCHA, M. A. **Cobertura: Uma métrica para aferir testes.** 2020. <<http://www.matera.com/blog/post/cobertura-uma-metrica-para-aferir-testes>>. [Online; acessado em 18 fev. 2020].
- ROUSE, M. **Web application (Web app).** 2020. <<https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>>. [Online; acessado em 21 jan. 2020].
- SALESFORCE. **About Heroku.** 2020. <<https://www.heroku.com/about>>. [Online; acessado em 20 jan. 2020].
- SIGNIFICADOSBR. **Significado de Conhecimento.** 2020. <<https://www.significadosbr.com.br/conhecimento>>. [Online; acessado em 22 jan. 2020].
- SILVA, A. **TDD: Na prática.** 2020. <<http://blog.redspark.io/tdd-na-pratica/>>. [Online; acessado em 20 jan. 2020].
- SOMMERVILLE, I. Software Engineering. Pearson Education, 2015. ISBN 9780133943276. Disponível em: <https://books.google.com.br/books?id=x_qZCgAAQBAJ>.
- SPRYMEDIA LTD. **DataTables - Manual.** 2020. <<https://datatables.net/manual/index>>. [Online; acessado em 18 jan. 2020].
- THE JQUERY FOUNDATION. **How jQuery Works.** 2020. <<https://learn.jquery.com/about-jquery/how-jquery-works/>>. [Online; acessado em 20 jan. 2020].
- THE PHP GROUP. **O que é o PHP?** 2020. <https://secure.php.net/manual/pt_BR/intro-what-is.php>. [Online; acessado em 05 nov. 2019].
- WIKILIVROS. **Redes de computadores/Camada de aplicação.** 2020. <https://pt.wikibooks.org/wiki/Redes_de_computadores/Camada_de_aplicação>. [Online; acessado em 20 jan. 2020].

Apêndices

A. Exemplos de Casos de Teste do LaraHelp

Neste apêndice são apresentadas as Listagens 3 à 5 que ilustram respectivamente casos de teste de funcionalidade, navegador e unitário do LaraHelp.

Listagem 3. Caso de teste de funcionalidade do LaraHelp.

```

1 /** @test */
2 public function it_authenticates_and_redirects_user()3 {
4     $user = factory(User::class)->create();
5
6     $response = $this->post(route('login'), [
7         'email' => $user->email,
8         'password' => 'password'
9     ]);
10
11     $response->assertRedirect(route('admin.dashboard'));
12     $this->assertAuthenticatedAs($user);
13}

```

Listagem 4. Caso de teste de navegador do LaraHelp.

```

1 /** @test */
2 public function it_can_search_on_home_page_and_redirects_to_articles()3 {
4     $article = factory(Article::class)->create();
5
6     $this->browse(function (Browser $browser) use ($article) {
7         $browser->visit('/')
8         ->type('search', $article->title)
9         ->press('execute_search')
10        ->assertSee('Artigos')
11        ->assertSee($article->title)
12        ->assertSee($article->summary)
13        ->assertInputValue('search', $article->title);
14    });
15}

```

Listagem 5. Caso de teste unitário do LaraHelp.

```

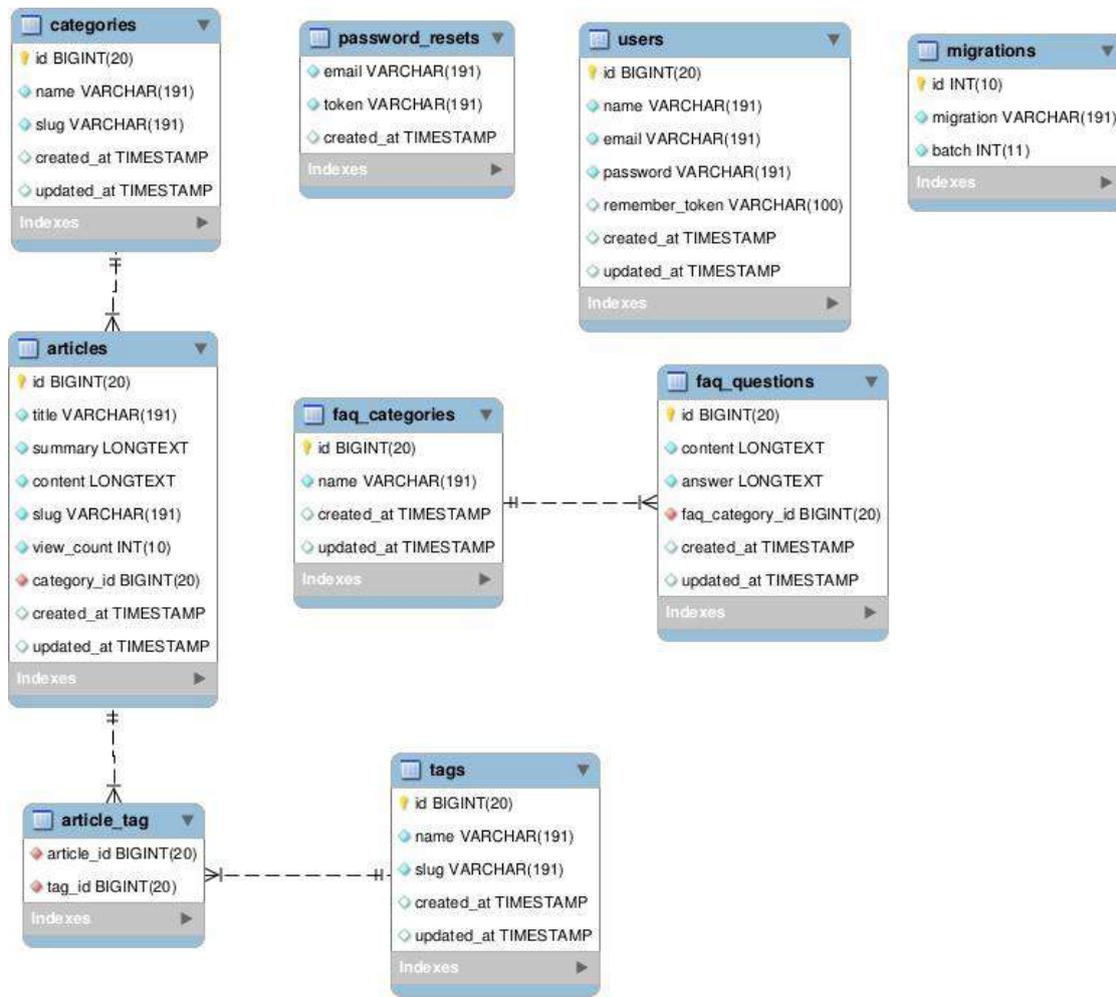
1 class SlugRuleTest extends TestCase2 {
2     protected $rule;
3     public function setUp(): void {
4         parent::setUp();
5         $this->rule = new SlugRule();
6     }
7     ...
8     /**
9      * Check that the invalid slugs does not pass
10    *
11    * @dataProvider invalidSlugs
12    * @param string $slug
13    * @return void
14    * @test
15
16    */
17    public function it_rejects_invalid_slugs($slug)
18    {
19        $this->assertEquals(0, $this->rule->passes('test', $slug));
20    }
21
22    public function invalidSlugs()
23    {
24        return [
25            [ '-user -name' ],
26            [ 'user --name' ],
27            [ 'username - ' ]
28        ];
29    }
30    ...
31 }

```

B. Diagrama Entidade-Relacionamento do LaraHelp

Neste apêndice é apresentado a Figura 11 que representa o DER do LaraHelp.

Figura 11. Diagrama Entidade-Relacionamento do LaraHelp.



C. Exemplos de Páginas do LaraHelp

Neste apêndice são apresentados exemplos de páginas do LaraHelp, ilustradas nas Figuras 12 à 24.



LaraHelp Home Artigos Categorias Tags Perguntas Frequentes (FAQ) Admin

Olá, como podemos ajudar?

Buscar

Categorias Populares: dolor, et, nisi

Categorias de Artigos

 quibusdam 1 artigo(s).	 laboriosam 1 artigo(s).
 veniam 1 artigo(s).	 asperiores 1 artigo(s).
 magni 1 artigo(s).	 libero 1 artigo(s).

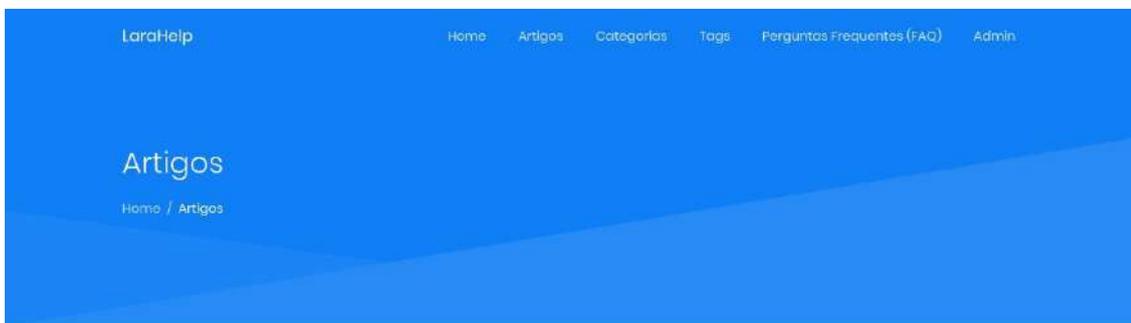
Artigos populares

Comece aqui para encontrar as melhores respostas possíveis de nossos especialistas.

Placeat quae nulla eos ullam voluptate animi fugit.	19/01/2020 16:48:35
Magni temporibus odit maiores recusandae aut et.	19/01/2020 16:48:35
Sint quia expedita aliquam aliquid sunt quis sint.	19/01/2020 16:48:35

[Ver Todos os Artigos >](#)

Figura 12. Página inicial do LaraHelp.



Buscar

Categoria:

Tags:

Buscar

Saepe et autem quas enim repellendus sit.

Categoria: repellendus

Vellit quis consequatur reiciendis minus quasi eos. Et nihil qui qui nihil reprehenderit noscunt sunt.

Postado em: 19/01/2020 16:48:34 Última atualização: 29/02/2020 10:53:10

Tags: Nenhuma. [Ler Mais](#)

Quibusdam dolore queraat nobis asperiores quo placeat.

Categoria: quibusdam

Repudiandae ut harum beatae minima maxima.

Postado em: 19/01/2020 16:48:34 Última atualização: 19/01/2020 16:48:34

Tags: enim [Ler Mais](#)

Et aut rerum odit officii.

Categoria: veniam

Sed quod facere sequi. Id veritatis et est ad dicta in itaque.

Postado em: 19/01/2020 16:48:34 Última atualização: 19/01/2020 16:48:34

Tags: enim esse enim [Ler Mais](#)

Artigos Populares

- Placeat quiae nulla eos ullam voluptate animi fugit.
- Magni temporibus odit maiores recusandae aut et.
- Sint quia expedita aliquam aliquid sunt quis sint.

Últimos Artigos

- Iusto eum cupiditate totam sunt.
- Vel velit que omnis harum.
- Vel duimus recusandae et et dolores.

Tags Populares

enim esse enim

ISSN – 1808-9305



VOOS

REVISTA POLIDISCIPLINAR ELETRÔNICA

Figura 13. Página de pesquisa de artigos do LaraHelp.

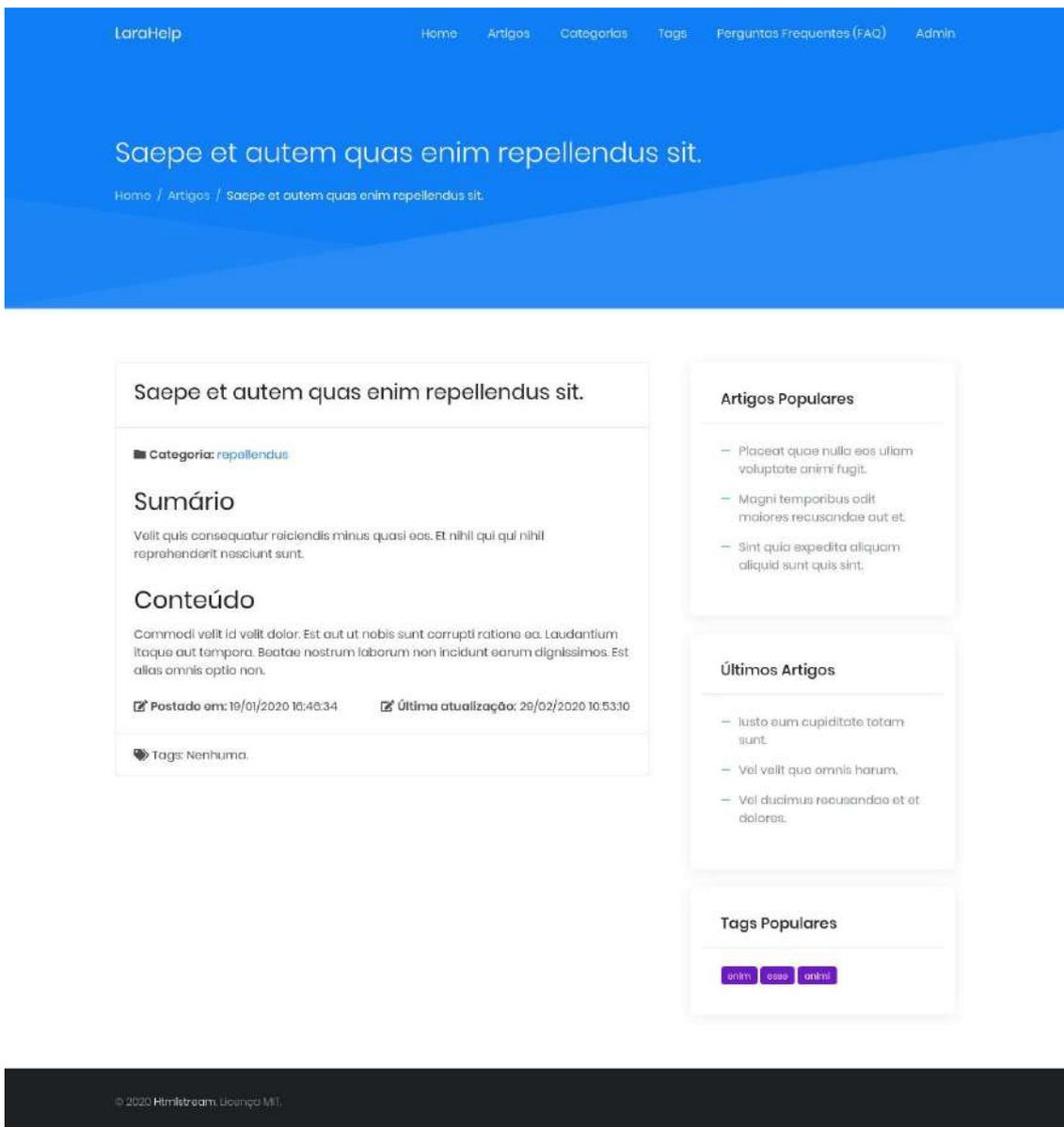


Figura 14. Página de visualização de um artigo do LaraHelp.

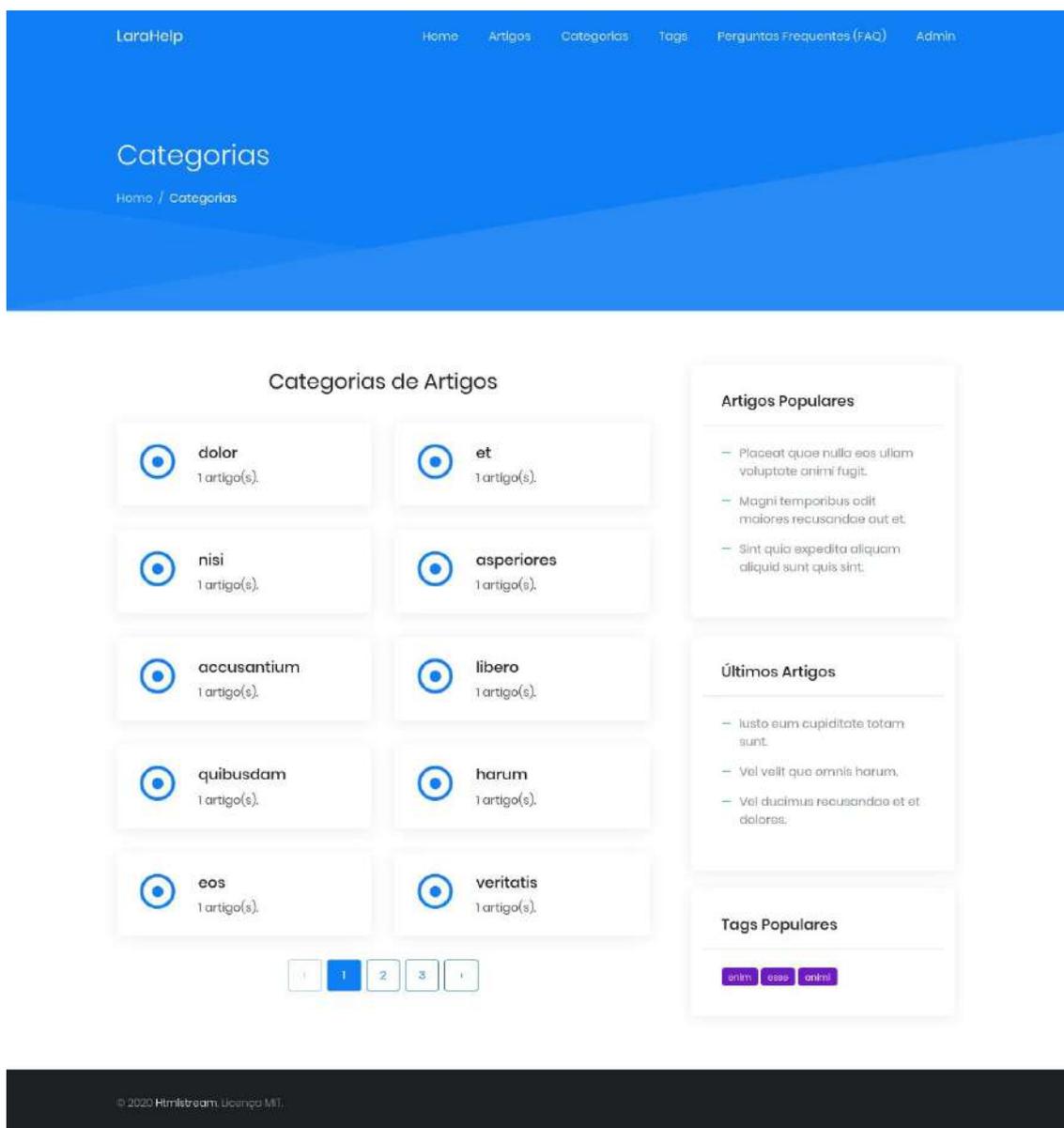


Figura 15. Página de categorias de artigos do LaraHelp.

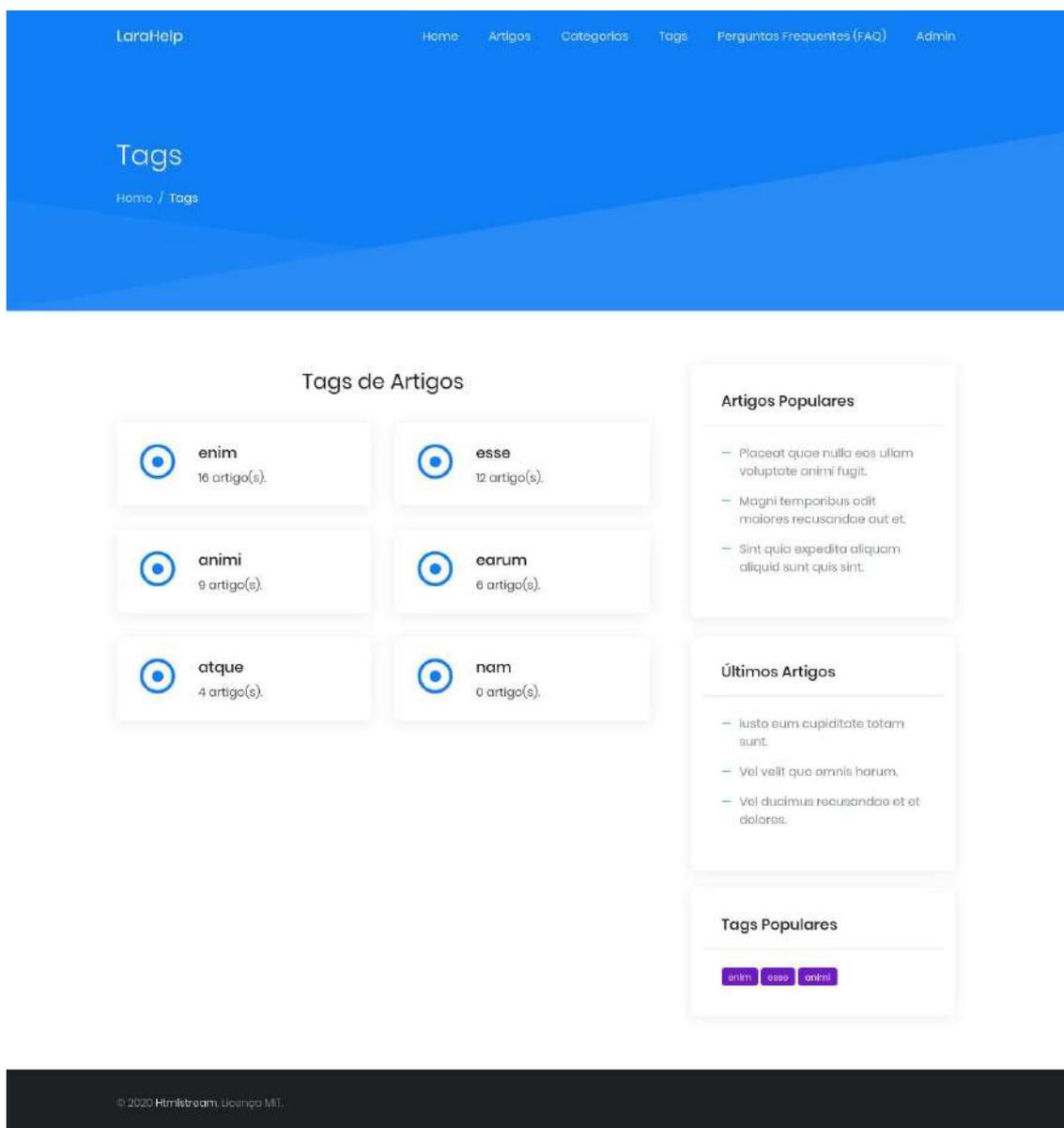


Figura 16. Página de *tags* de artigos do LaraHelp.

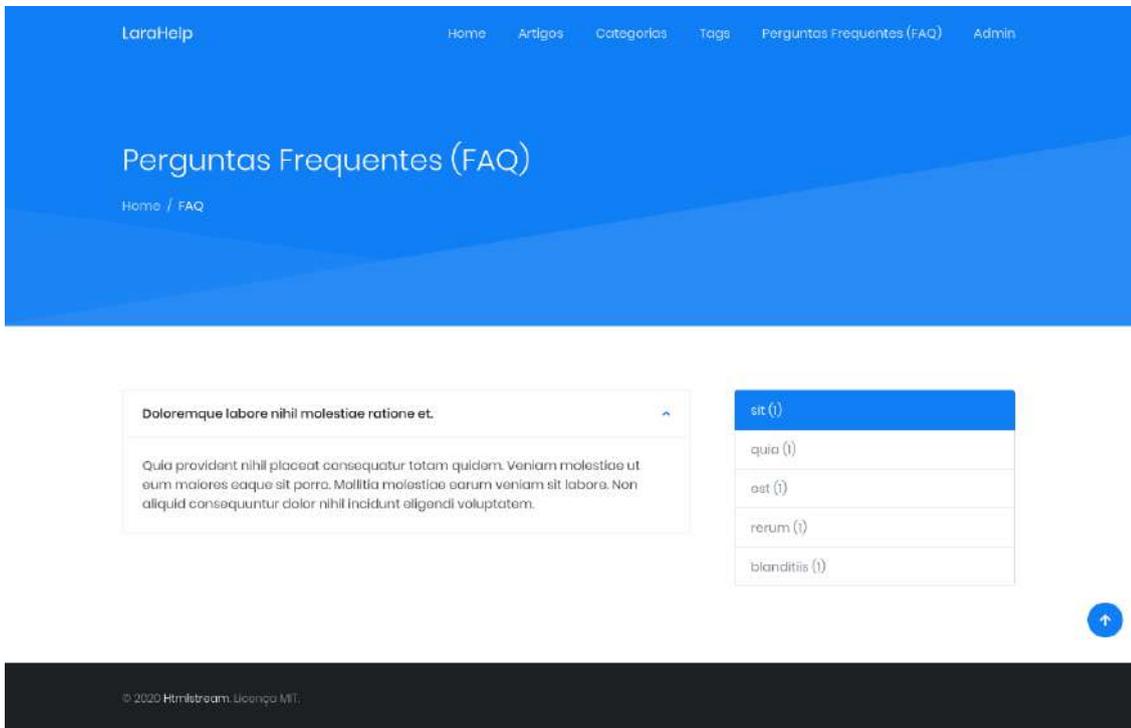
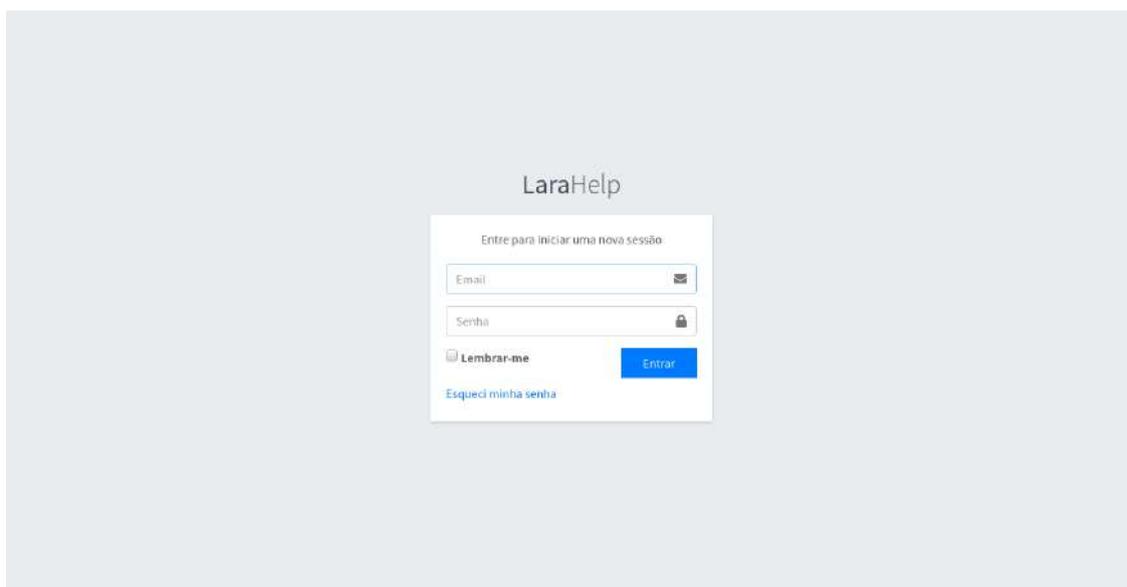


Figura 17. Página de perguntas frequentes do LaraHelp.

Figura 18. Página de login no painel administrativo do LaraHelp.



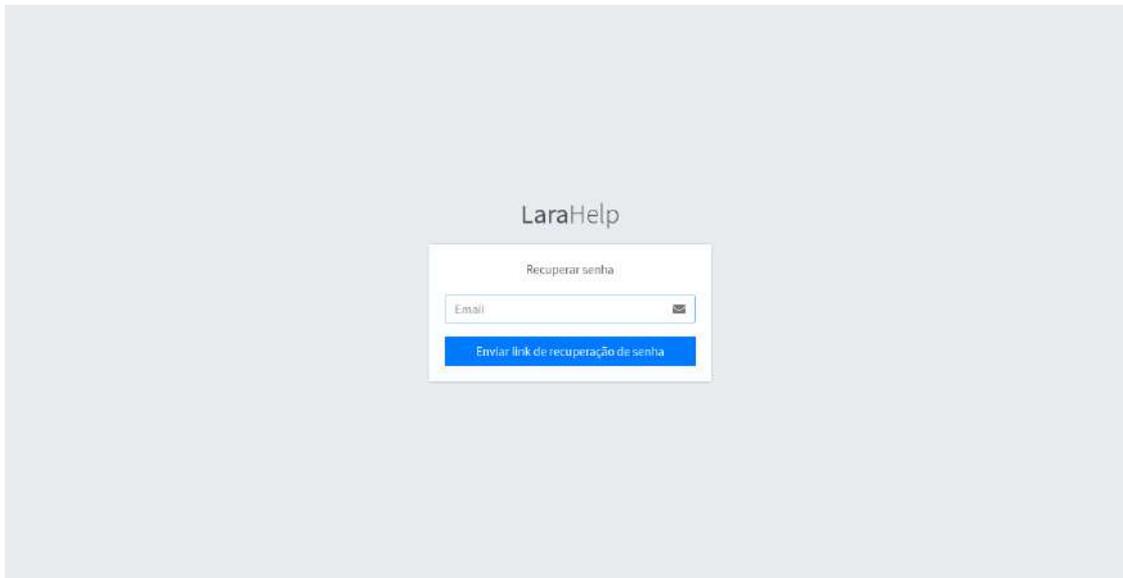


Figura 19. Página de resetar senha do LaraHelp.

LaraHelp Saír

- Dashboard
- Usuários
- Categorias
- Tags
- Artigos
- Perguntas Frequentes

Dashboard

Artigos
20

#	Artigo	Visualizações
21	Rerum quidem quibusdam dolores.	4930
101	Rem at ducimus delectus et fuga doloribus.	4897
31	Dicta illo consequatur cupiditate dolore.	4746
91	Quasi illum excepturi alias fugit labore.	4620
121	Qui beatae expedita sint et corrupti.	3901
151	Qui deserunt et omnis a nostrum.	3649

Categorias
25

#	Categoria	Total de Visualizações
71	maxime	4930
151	doloribus	4897
81	mollitia	4746
141	quis	4620
171	sit	3901
201	dignissimos	3649
91	voluptatibus	3380
111	ex	2986
51	eos	2823
231	iste	2117

Tags
6

#	Tag	Total de Visualizações
51	dolorem	45051
41	cum	34302
21	et	32347
31	voluptatem	32347
11	eius	27864
1	dolor	13411

Figura 20. Página de *dashboard* administrativo do LaraHelp.

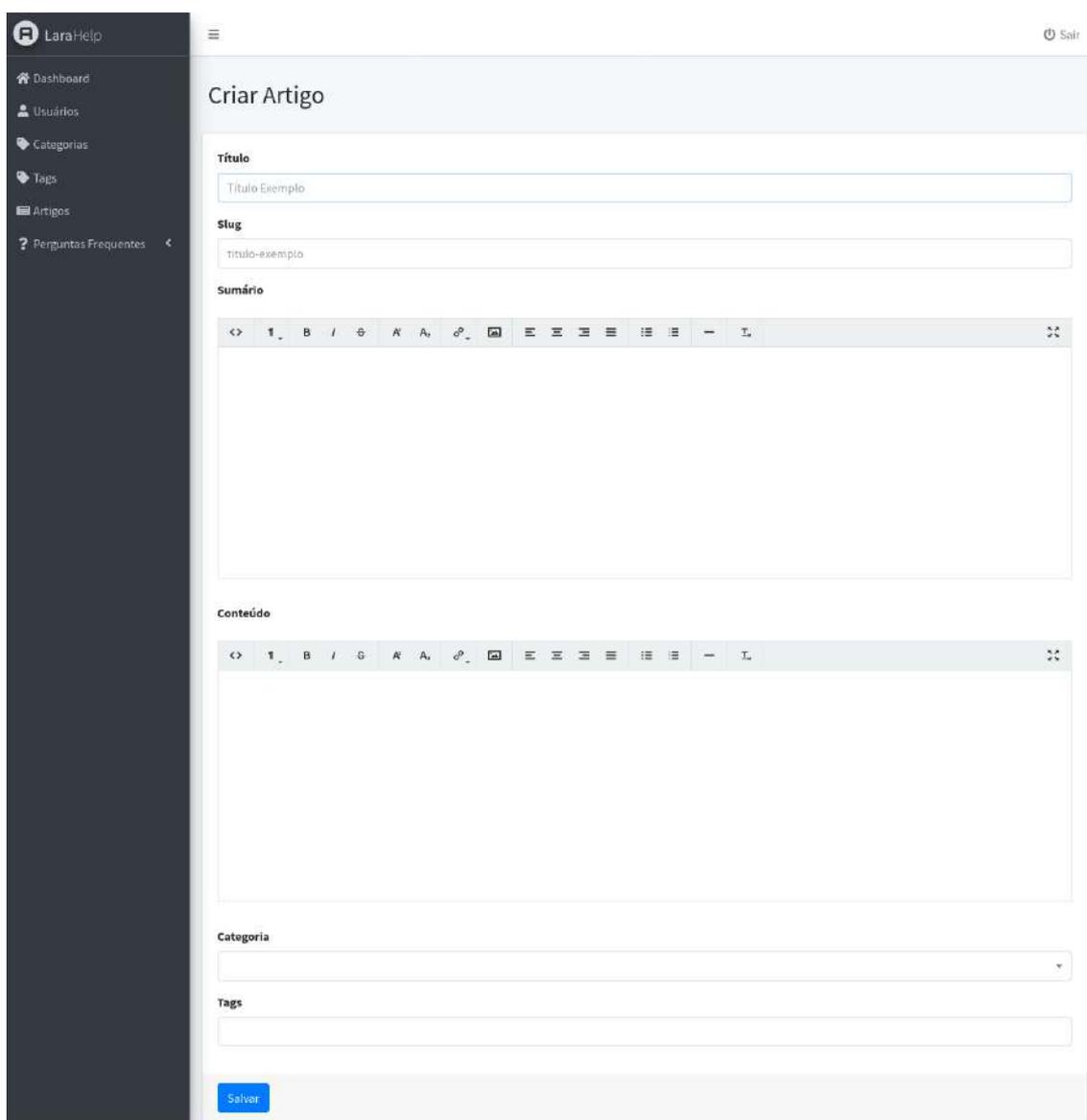


Figura 21. Página de criar um artigo do LaraHelp.

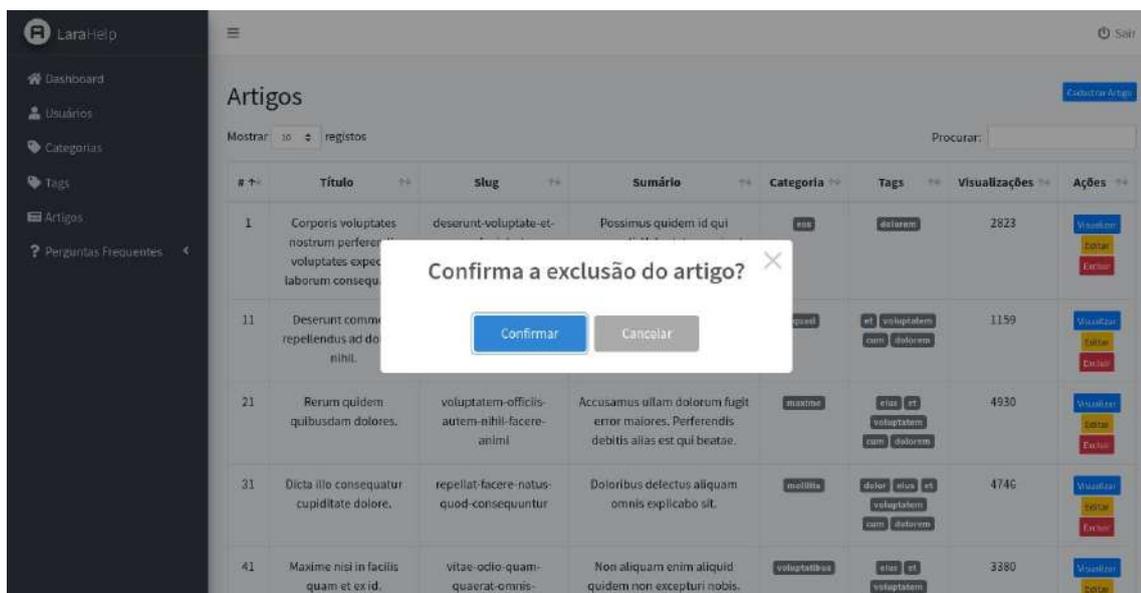
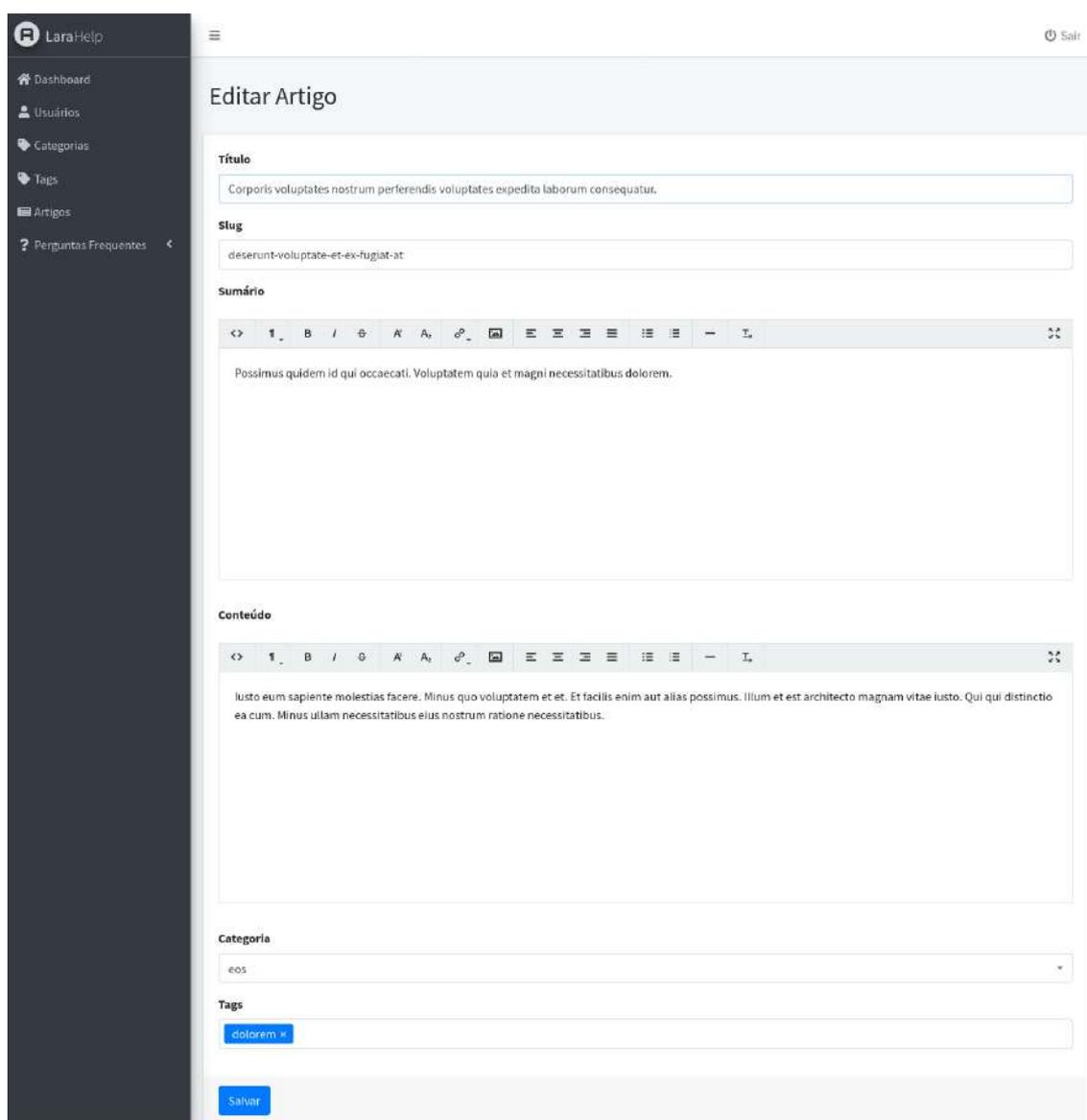


Figura 22. Modal de excluir um artigo do LaraHelp.



LaraHelp Sair

Dashboard
Usuários
Categorias
Tags
Artigos
Perguntas Frequentes

Editar Artigo

Título

Corporis voluptates nostrum perferendis voluptates expedita laborum consequatur.

Slug

deserunt-voluptate-et-ex-fugiat-at

Sumário

Possimus quidem id qui occaecati. Voluptatem quia et magni necessitatibus dolorem.

Conteúdo

Iusto eum sapiente molestias facere. Minus quo voluptatem et et. Et facilis enim aut alias possimus. Illum et est architecto magnam vitae iusto. Qui qui distinctio ea cum. Minus ullam necessitatibus eius nostrum ratione necessitatibus.

Categoria

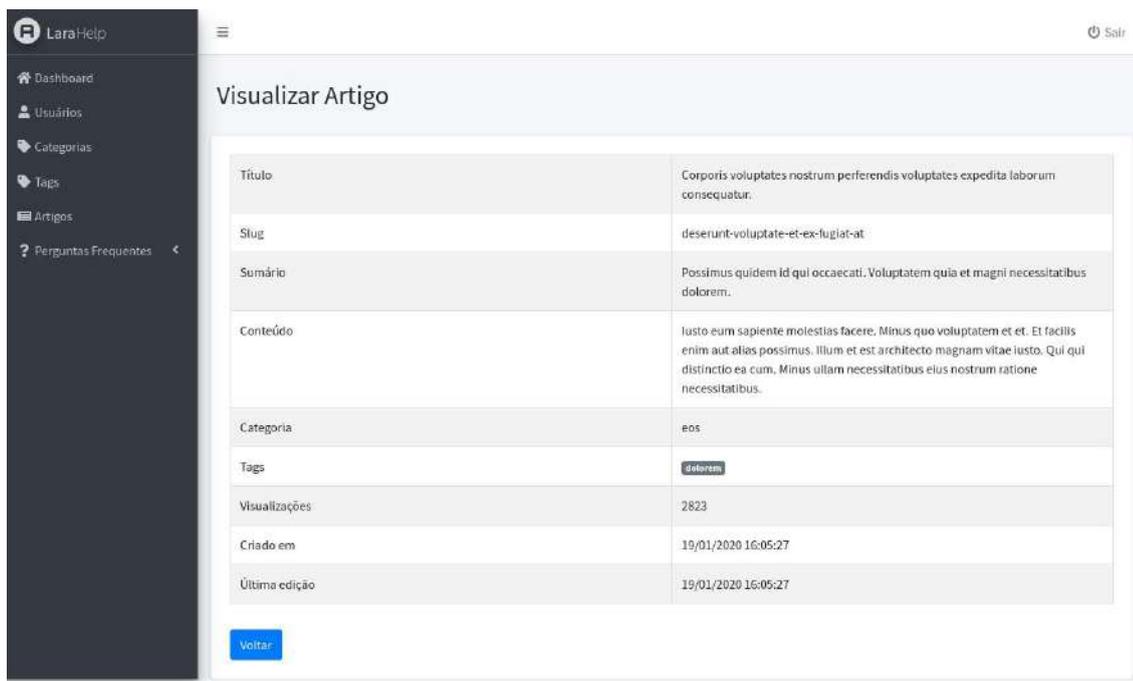
eos

Tags

colorem

Salvar

Figura 23. Página de editar um artigo do LaraHelp.



LaraHelp

Sair

Visualizar Artigo

Título	Corporis voluptates nostrum perferendis voluptates expedita laborum consequatur.
Slug	deserunt-voluptate-et-ex-lugiat-at
Sumário	Possimus quidem id qui occaecati. Voluptatem quia et magni necessitatibus dolorem.
Conteúdo	Iusto eum sapiente molestias facere. Minus quo voluptatem et et. Et facilis enim aut alias possimus. Illum et est architecto magnam vitae iusto. Qui qui distinctio ea cum. Minus ullam necessitatibus eius nostrum ratione necessitatibus.
Categoria	eos
Tags	dolorem
Visualizações	2823
Criado em	19/01/2020 16:05:27
Última edição	19/01/2020 16:05:27

[Voltar](#)

Figura 24. Página de visualizar um artigo (administrador) do LaraHelp.