



## UFSMOOC: uma plataforma institucional para cursos online gratuitos

Dorneles, Ana Paula M.<sup>1</sup>, Frigo, Augusto P.<sup>1</sup> Silva, Thiago R.<sup>1</sup>,  
Dorneles, Lucio S.<sup>2</sup>, Librelotto, Giovani R.<sup>1</sup>

<sup>1</sup>Curso de Ciência da Computação – Universidade Federal de Santa Maria  
Santa Maria – RS – Brazil

<sup>2</sup>Departamento de Física – Universidade Federal de Santa Maria  
Santa Maria – RS – Brazil

trsilva@inf.ufsm.br, anapaulamilitz@gmail.com, frigo.augusto@gmail.com,  
lucio.dorneles@ufsm.br, librelotto@inf.ufsm.br

**Abstract.** *Teaching options aimed at the community, at UFSM (Federal University of Santa Maria), could reach a greater number of people, if it weren't for the absence of a platform for the use of MOOCs (Massive Open Online Courses). Here, the creation of an institutional platform that meets this need is presented; with free courses, from all knowledge areas. The implementation weighed aspects of the system's usability and design, as well as its functionality and security. In this context, the developed software has shown the potential to become a relevant tool in the democratization of knowledge in Brazil.*

**Resumo.** *As opções de extensão focadas no ensino para a comunidade, na UFSM (Universidade Federal de Santa Maria), poderiam alcançar um número maior de pessoas, não fosse a ausência de uma plataforma para uso de MOOCs (Massive Open Online Courses). Este artigo apresenta a criação de uma plataforma institucional que supra essa necessidade; com cursos gratuitos, de todas as áreas do conhecimento. A implementação considerou aspectos da usabilidade e design do sistema, bem como sua funcionalidade e segurança. Neste contexto, o software desenvolvido apresentou potencialidade de tornar-se uma ferramenta relevante na democratização do conhecimento no Brasil.*

### 1. Introdução

Cursos massivos abertos online (MOOC) são ferramentas, de longo alcance, que podem abordar diversas áreas do conhecimento e têm, como intuito original, proporcionar acesso livre ao conhecimento por meio de cursos, de curta duração, pela internet. Neste âmbito, são muito utilizados com o intuito de obtenção de certificação profissional ou aprendizado de novas competências por parte dos usuários. A UFSM, entretanto, não dispõe de uma plataforma institucional específica para a realização de MOOCs, dificultando, assim, a disseminação do conhecimento por esse meio, apesar do esforço de grupos, como o Programa de Educação Tutorial (PET) [1], mostrar-se visível no que tange a disseminação de conhecimento por meio de cursos.

A UFSMOOC, plataforma desenvolvida por estudantes e professores da UFSM, apresenta a proposta de criação de uma plataforma para ministrar e participar de MOOCs. Desse modo, a ferramenta propõe-se a prover uma plataforma acessível e focada em levar aos estudantes da universidade a experiência da docência em cursos

remotos, bem como o conhecimento advindo dessa docência à comunidade em geral. O *software* é construído com base na liberdade de escolha do ministrante - isto é - várias características do curso, como avaliação automatizada ou manual e existência de tutoria, ou não, são definidas por ele.

Este artigo está organizado da seguinte forma: na segunda seção apresenta-se a motivação da criação do sistema; a terceira seção descreve uma revisão dos sistemas existentes e similares; a quarta seção, por sua vez, mostra a construção do sistema proposto; na quinta tem-se os resultados obtidos e, por fim, a conclusão discute os pontos principais obtidos com este trabalho.

## 2. Motivação e justificativa

A pandemia da COVID-19 forçou a adoção do ensino remoto em grande parte dos países, a possibilidade de ministrar cursos presencialmente acaba, portanto, restrita. Conforme dados da UNESCO [2], em 27/09/2021, cerca de 128 milhões de estudantes foram afetados pelo fechamento de escolas. Além disso, independentemente da pandemia, cursos presenciais possuem reuso de material limitado, dificultando a reaplicação. Desse modo, uma plataforma institucional para aplicação de cursos online mostra-se uma alternativa ao problema, evitando a contaminação dos ministrantes e participantes pela Covid-19 e fornecendo conteúdo de qualidade semelhante.

Conforme dados da pesquisa TIC Domicílios de 2019 [3], 74% da população brasileira com 10 anos ou mais possui acesso à internet. Desse modo, a plataforma fica acessível a grande parte da população, já que dispõe de interface desktop e mobile. Ainda que alguns usuários possuem *hardware* antigo, o sistema é otimizado para o seu bom funcionamento nesses dispositivos, sem utilizar-se de mecanismos como anúncios.

O Ministério da Educação, no artigo 4º da RESOLUÇÃO Nº 7, DE 18 DE DEZEMBRO DE 2018 [4], institui que “As atividades de extensão devem compor, no mínimo, 10% (dez por cento) do total da carga horária curricular estudantil dos cursos de graduação, as quais deverão fazer parte da matriz curricular dos cursos”. A plataforma mostra-se, portanto, uma possibilidade de cumprimento dessa carga horária aos alunos necessitados.

## 3. Trabalhos relacionados

Diversos trabalhos científicos foram realizados para o estudo de *Massive Open Online Courses* (MOOCs), desde os seus surgimentos. Um dos primeiros projetos do gênero, o *MIT OpenCourseWare* [5], mostrou surpreendentes resultados em relação às metas iniciais estabelecidas por seus idealizadores. Seu sucesso foi tão grande que chegou a ser chamado de o Big Bang no Universo do Conhecimento, receber diversas ofertas de tradução para outras linguagens, fazer acordos com 900 universidades ao redor do mundo e atingir suas metas de publicação com custos menores do que o projetado, conforme explica Hal Abelson - um de seus dirigentes - no artigo “*The Creation of OpenCourseWare at MIT*” [6].

O uso de MOOCs também é realizado de maneira comercial por algumas das maiores plataformas no ramo, como a Coursera, EdX e Udacity [7]. Seus principais

componentes dos modelos de negócios incluem: segmento de cliente, valor proposição, canais de comunicação, fluxos de receita e estrutura de custos, conforme explicita o artigo Cornejo-Velazquez, E., et al., 2020 [8]. Suas estruturas de negócio também permitem geração de lucro por meio da parceria com universidades renomadas, como Harvard, Stanford e Yale - que, por sua vez, possuem suas próprias plataformas MOOC.

No Brasil, grande parte das Instituições de Ensino (IES) adeptas de MOOC originaram suas plataformas a partir de códigos-fonte abertos, em especial, os Institutos Federais pelo uso da Plataforma TIMTec [9]. No entanto, a maior parte das universidades brasileiras não possuem uma plataforma com tal finalidade e o processo de integração deste tipo de ensino na educação brasileira encontra-se em fase experimental, como apresentado no estudo.

Por tratar-se de uma plataforma advinda de uma IES, o UFSSMOOC distancia-se, em maior grau, de plataformas com fins lucrativos e objetiva, ao invés, acesso ilimitado aos cursos ofertados, com todos seus certificados disponibilizados de maneira gratuita. Por esta razão, o projeto possui maiores semelhanças com a plataforma pioneira, MIT OpenCourseWare, e com as instituições brasileiras supracitadas. Seu diferencial, no entanto, vem a ser a possibilidade de colaboração por parte de grupos do Programa de Educação Tutorial (PET) da Universidade Federal de Santa Maria.

O oferecimento de cursos síncronos/graduais, provindo dos PETs, traz a viabilidade de ganhos de Atividades Complementares de Graduação (ACG) para os integrantes destes grupos. Ademais, permite que alunos matriculados em alguns destes cursos tenham, como benefício, suas atividades corrigidas por ministrantes que optem pela aplicação de avaliações não-automatizadas. Não exclui-se, entretanto, a possibilidade de participantes dos grupos PET ministrarem cursos assíncronos e abertos à matrícula a qualquer momento, utilizando recursos como correção automática para facilitar a avaliação.

## **4. Metodologia**

Esta seção descreve um estudo referente ao desenvolvimento do sistema de cursos remotos ministrados por docentes e discentes da UFSSM, tendo como público-alvo a comunidade em geral.

### **4.1. Desenvolvimento do sistema**

O *software* foi projetado, primeiramente, por meio da análise do sistema. Foram realizados: Listagem de requisitos funcionais e não funcionais, descritivo de casos de uso, diagrama de casos de uso, diagrama de classes, modelo de entidade relacionamento. A prototipação do Design foi realizada por meio da ferramenta Figma [10].

O sistema foi desenvolvido utilizando as linguagens HTML [11], CSS [11], Javascript [12], PHP [13] e MySQL [14], bem como os *frameworks* Laravel [15] e Vue [16]. Destaca-se a utilização de três principais linhas de desenvolvimento: *design*, *front-end* e *back-end*.

#### **4.1.1. Planejamento do projeto**

O planejamento do projeto foi feito mediante reunião dos integrantes, definindo quais seriam os principais recursos e funcionalidades demandados. Foi definido, portanto, que seria feita primeiramente a análise do sistema, para posteriormente desenvolvê-lo propriamente.

#### 4.1.2. Análise do Sistema

Inicialmente, foi realizada a criação de um documento que englobou diversas informações necessárias ao desenvolvimento do projeto, incluindo a definição dos tipos de usuário, suas responsabilidades e suas possíveis ações na plataforma. Então, destas descrições mais genéricas, originaram-se os requisitos funcionais e não funcionais do sistema, conforme exemplifica a Figura 1. A partir deles, foram extraídos e listados os possíveis casos de uso.

Posteriormente, foram descritas as informações consideradas relevantes para o futuro desenvolvimento do sistema para cada caso de uso. Deste modo, pode-se citar, concepções como o tipo de usuário envolvido, uma descrição específica, os possíveis fluxos de interação usuário-sistema, restrições, dados envolvidos e possibilidade de o caso de uso encaminhar a outro ou possuir um outro como pré-requisito.

Assim, partindo destas descrições mais detalhadas, foi desenvolvido um diagrama de casos de uso, segundo demonstrado na Figura 2. Buscou-se apresentar, de forma visualmente organizada e intuitiva, as interações de cada usuário com cada caso de uso em si, bem como a ordem de ocorrência dos mesmos. Também, foi desenvolvido um diagrama de classes para auxiliar no processo de maneira complementar.

Por fim, foi desenvolvido um modelo entidade relacionamento como mostrado na Figura 3. Desta maneira, um protótipo do banco de dados permitiu sua visualização e organização de maneira mais prática e simples, com análises mais adequadas e implementação de alterações anteriores à fase de programação.

##### RF14 – Contatar Tutor

Caso a tutoria seja fornecida no curso, o sistema deve conferir ao aluno o direito de contatar o tutor para que suas dúvidas sejam esclarecidas.

##### RF15 – Solicitar Certificado

Caso o curso ofereça certificação, deve, também, possibilitar ao aluno a solicitação de emissão do certificado e o recebimento deste.

##### RF16– Abandonar Curso

O aluno pode abandonar um curso em andamento, se desejar. Deve ser informado das consequências e deve explicitar o porquê do abandono.

##### RF17 –Recuperar de Senha

O usuário deve poder solicitar uma recuperação de senha. Um e-mail é enviado para seu endereço de e-mail cadastrado.

Figura 1. Listagem de requisitos funcionais e não funcionais



Figura 2. Imagem do Diagrama de Casos Uso.

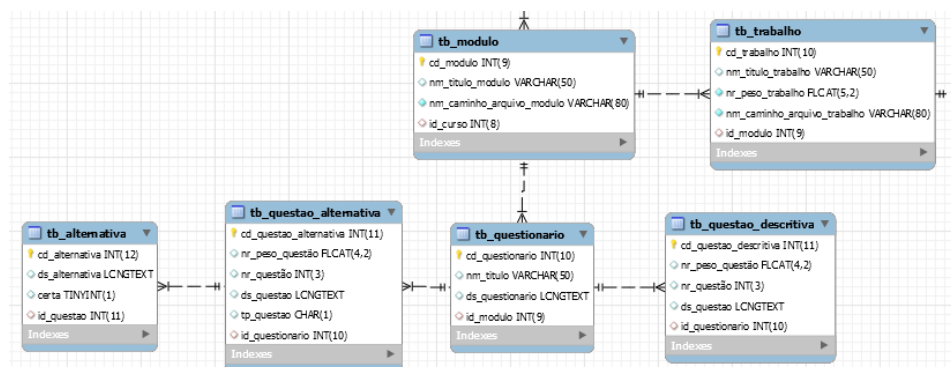


Figura 3. Modelo entidade relacionamento

## 4.2. Design

O processo envolveu o desenvolvimento de um sistema de design, um sitemap, a criação dos frames e wireframes, e a prototipação final das páginas. Além disso, foram utilizadas, como principal base teórica para a elaboração dos componentes do sistema, as dez heurísticas de Nielsen [17].

A primeira etapa do design envolveu a elaboração de *wireframes*: esboços simples da estrutura visual do site que deram forma à ideia do projeto. Por meio da utilização do *software* Balsamiq [18], nos wireframes de todas as páginas principais foram exibidos os elementos essenciais da interface. Algumas destas páginas criadas foram a *landing page*, a página de visualização de um curso e a página de visualização de categorias de cursos.

Posteriormente, um sistema de design foi estabelecido para a padronização e consistência da plataforma. No sistema, foram incluídos todos os ícones e figuras criadas para o projeto, tal como, um sistema de cores contendo as cores primárias, as de alerta e as sombras. O azul, da paleta de cores da UFSM, serviu como base na criação das demais cores, por meio da mudança da luminosidade de seus formatos HSL(Hue-Saturation-Brightness) [19].

Na terceira etapa, a plataforma Miro [20] foi utilizada para a criação de dois sitemaps: o primeiro com a estrutura do site do ponto de vista do professor, e o segundo do ponto de vista do aluno. Assim, enquanto o segundo sitemap previu páginas como a de visualização de cursos em andamento por parte do aluno, o primeiro conteve adicionais que permitiam, também, a visualização de seus cursos ministrados e arquivados. Em vista disso, outras ações autorizadas aos professores - como a adição de cursos à plataforma e a edição de conteúdo dos mesmos - foram restringidas aos alunos.

Ademais, as dez heurísticas de Nielsen serviram como base para todo processo de design. A utilização das mesmas pode ser observada em diversas partes do projeto, dentre as quais, é possível citar: primeira heurística no fornecimento de *feedback* sobre o envio do e-mail para redefinição de senha; segunda heurística nos ícones que assemelham-se a elementos não-virtuais e cotidianos ao usuário; terceira heurística na possibilidade de cancelamento do envio de um trabalho avaliativo; quarta heurística no arredondamento dos botões consistente em todas as páginas; quinta heurística no pedido

de confirmação ao aluno sobre o envio de seu trabalho; sexta e sétima heurísticas nas alternativas de login pelas redes sociais, como maneira de reduzir a quantidade de informações que o usuário necessita lembrar e de flexibilizar as opções de uso de login existentes; oitava heurística nas configurações, deixadas em um botão secundário do menu; nona heurística na sugestão de solução do esquecimento da senha e décima heurística na criação de uma página de dúvidas frequentes.

Por fim, utilizando a plataforma Figma, cinquenta e quatro *frames* de design - equivalentes a páginas Web - foram criados; representando as funcionalidades do sistema e tendo, seus conteúdos, variado entre ações do aluno e ações do professor, conforme exemplificam as Figura 4 e Figura 5. Ainda nesta ferramenta, a prototipação do projeto foi realizada para a apresentação deste nas reuniões em grupo e para a verificação da existência de todas as páginas necessárias na implementação do site.

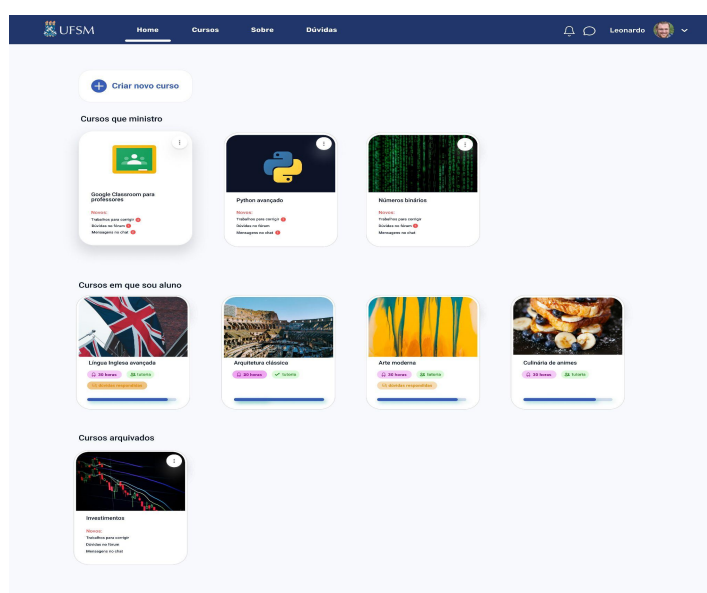


Figura 4. Design da página de gerenciamento de cursos do professor



Figura 5. Design da página de dúvidas frequentes

### 4.3. Front-end

No front-end, ocorreu a adaptação do design para, primeiramente, o formato de arquivo “.blade.php”. Foram adaptadas as funcionalidades referentes ao que será executado no dispositivo do usuário, bem como a comunicação com o *back-end*. Posteriormente, ocorreu a migração para o formato “SPA” (*single page application*), com a utilização do *framework* Vue. Na Figura 6, é explicitada a adaptação feita da página de “Dúvidas Frequentes”, projetada no design, para o formato final.

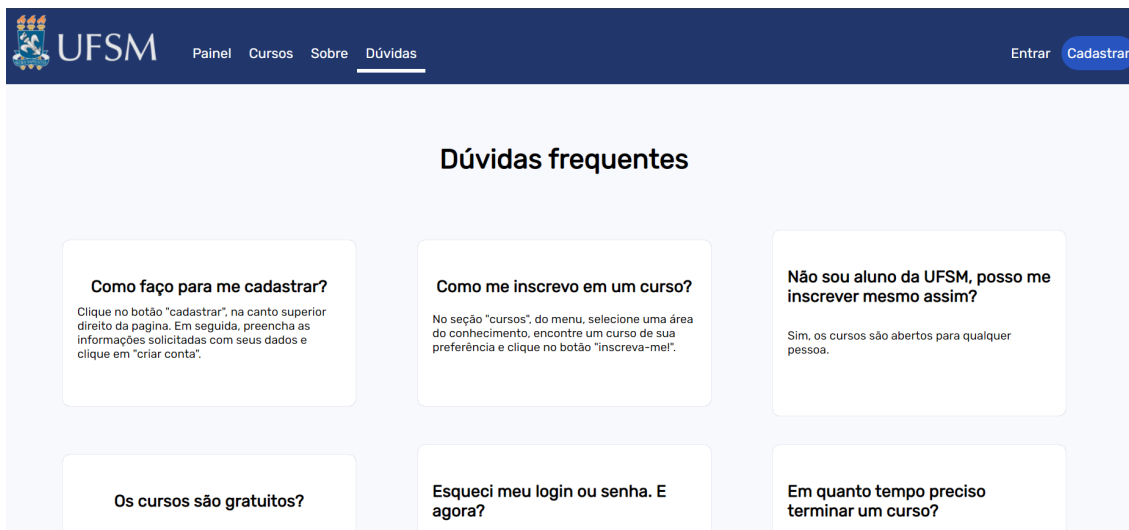


Figura 6. Adaptação da página “Dúvidas frequentes” para o front-end

### 4.4. Back-end

No *back-end*, foi feita a adaptação do *server-side*, com as funcionalidades a serem executadas pelo servidor, tais como autenticação e acesso ao banco de dados, por meio das linguagens PHP e MySQL, sendo, como *framework*, utilizado o Laravel 7 e, posteriormente, feito o *upgrade* para a versão 8.

#### 4.4.1. O uso do PHP no back-end

A linguagem de programação PHP foi amplamente utilizada dentro do contexto de processamento de informações enviadas pelo usuário e parte funcional do servidor, bem como a comunicação com o banco de dados MySQL. Sua utilização se deu no contexto do *framework* Laravel, sendo utilizado, principalmente, para os *Controllers* e *Middlewares*.

Os *controllers* são responsáveis pela parte lógica interna do sistema. São utilizados para várias funções, como criação de usuários, criação de novos cursos e retorno de cursos disponíveis.

Os *Middlewares* foram utilizados para restrição de acesso em determinadas páginas. Não é desejado que um usuário comum acesse a página de administrador, por exemplo. Por isso, é feita a verificação de tipo de usuário antes de continuar. Conforme explicitado na Figura 7, o usuário não pode prosseguir sem estar autenticado e possuir o

usuário tipo 2, ou seja, administrador. Caso uma dessas duas condições não seja cumprida, o usuário será redirecionado para a página inicial com erros.

```
public function handle($request, Closure $next)
{
    if (Auth::user() && Auth::user()->type_user == 2){
        return $next($request);
    }
    else{
        return redirect('home')->with('error','Voce nao tem acesso de administrador.');
```

Figura 7. Middleware que verifica se o usuário é administrador

#### 4.4.2. Banco de dados do back-end

O banco de dados foi construído por meio do *Database Query Builder*, ferramenta proveniente do *framework* Laravel. Utilizou-se como alvo do *Query Builder* a tecnologia MySQL.

Destaca-se que, com a ferramenta de construção utilizada, é possível migrar para outra tecnologia facilmente, assim como reconstruir o banco de dados com a mesma tecnologia. Facilita-se, assim, os esforços de refatoração, quando necessários.

Na Figura 8, é mostrada a função “up\*()”, usada para “erguer” o banco de dados. Nesta função, há vários exemplos de campos a serem construídos, como os campos “id”, “CPF” e e-mail (chaves únicas) e campos que podem repetir, como cidade e tipo de usuário.

```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('surname');
        $table->char('CPF', 11)->unique();
        $table->string('email', 40)->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->char('UF', 2);
        $table->string('city');
        $table->string('password');
        $table->char('type_user', 1);
        $table->rememberToken();
        $table->timestamps();
        $table->text('description')->nullable();
    });
}
```

Figura 8. Criação do banco de dados por meio do *Laravel Query Builder*

#### 4.4.3. Segurança do sistema

A fim de garantir a segurança e integridade dos usuários e do sistema, foram tomadas medidas para bloquear ataques cibernéticos.

Para evitar ataques do tipo SQL Injection [21], todas as entradas do banco de dados são filtradas antes de serem inseridas, no banco de dados intermediado pelo



*Laravel*. Assim, não é possível inserir trechos que possam levar à execução arbitrária do acesso ao banco de dados, como os caracteres "--", que poderiam servir para tornar um trecho do código executado um comentário, burlando, por exemplo, verificações de senha.

Destaca-se, também, a filtragem de *script* [22] do *framework* Vue, que executa a realocação de caracteres-chave. Assim, um usuário nomeado "<script>", por exemplo, consegue se registrar na aplicação com esse nome. Os caracteres "<" e ">", entretanto, são trocados por equivalentes não reconhecidos como *tags* na linguagem HTML. Desse modo, a injeção de uma *string* contendo código HTML não é problemática, pois o código não será executado.

Por fim, é importante destacar o método seguro de armazenamento de senhas. Quando as senhas de usuário chegam ao servidor, estas passam pela função *Bcrypt* [23], que utiliza um método do tipo *hash* [24]. Esse método criptografa a senha em uma via (ou seja, não é possível desfazer a criptografia) antes de armazená-la no banco de dados, fazendo com que, mesmo em um eventual vazamento do banco de dados, não ocorra o comprometimento da senha do usuário.

## 5. Resultados parciais

O desenvolvimento do sistema está concluído e a fase atual é a de testes e correções de problemas. Até o momento, o sistema oferece diversas funcionalidades para os usuários principais, os ministrantes e os alunos. O usuário responsável por ministrar cursos consegue cadastrar os cursos, atribuir material à eles - sejam em vídeos, textos ou imagens - e trabalhos a serem entregues, além de ter um acesso a lista de alunos de cada curso que ministra.

Na versão atual do sistema, o usuário aluno pode: cadastrar-se e logar-se e na plataforma; listar e visualizar os cursos sendo ofertados, bem como inscrever-se nestes; acessar e visualizar os materiais disponíveis e realizar entrega de trabalhos de cursos em andamento e, por fim, ele pode solicitar a permissão para se tornar um ministrante.

O último tipo de usuário, é o administrador, responsável por gerenciar os demais usuários e algumas funcionalidades do sistema, como desativar cursos, desvincular alunos de cursos e avaliar as requisições dos alunos que desejam se tornar ministrantes.

Ademais, a segurança do *software* demonstrou-se eficiente nos testes realizados; após diversas tentativas de *SQL Injection*, não foi obtido sucesso na injeção de qualquer tipo de *script* invasivo. A utilização de meios adequados para a autenticação do usuário, os *middlewares*, também reforçaram a segurança referente ao acesso indevido de informações reservadas para tipos de usuário diferentes.

## 6. Conclusão

Este trabalho apresentou o UFSSMOOC, um sistema para gerenciamento de cursos remotos que possam ser utilizado por alunos e professores da instituição, uma alternativa para cumprir a obrigatoriedade da extensão em cursos de graduação no Brasil, de forma que os discentes possam levar o conhecimento produzido na universidade à comunidade em geral de uma forma mais adequada.

Graças à fase de análise, realizada com o uso de diversas informações fornecidas pelos orientadores, desenvolveu-se uma série de funcionalidades adaptadas ao modelo de curso ofertado comumente pela UFSM. Dentre estas, pode se citar, a possibilidade do aluno participar dentro do curso como um tutor ou gerente de atividades, resultando em um sistema mais adequado para a universidade. Outras etapas, como a construção de um modelo entidade relacionamento e a prototipação das telas, demonstraram-se essenciais para a discussão de ideias e descobrimento de possíveis problemas, permitindo que o *software* fosse modificado e blindado a fim de evitar esforços de refatoração.

A ferramenta também demonstra um alto potencial para amplificar a disseminação do conhecimento atualmente provida, pela universidade, de maneira local, tendo em vista sua simplicidade e seu alcance gerado pela identidade de um software próprio da UFSM.

Em vista disso, considerando as necessidades de um meio propício à dedicação de horas em projetos de extensão, por parte dos graduandos, e de uma plataforma online própria para a oferta de cursos à sociedade - esta última, inclusive, acentuada pela pandemia - o sistema demonstrou-se uma solução adequada.

O planejamento de trabalhos futuros envolve, inicialmente, a realização de objetivos da etapa de finalização, como o polimento, a procura e a correção de erros do sistema. Ademais, objetiva-se, posteriormente, a disponibilização da plataforma para o público, por meio da introdução de cursos que abranjam todas as áreas do conhecimento estipuladas pela UFSM. Por fim, a adaptação da responsividade do sistema para um melhor fornecimento de suporte aos dispositivos mobile, tal como o reajuste de detalhes de usabilidade com base no feedback dos usuários da plataforma.

## Referências Bibliográficas

- [1] Programa de Educação Tutorial. MEC. Disponível em: <<http://portal.mec.gov.br/component/tags/tag/programa-de-educacao-tutorial>>. Acesso em: 29 set. 2021.
- [2] Education: From disruption to recovery. Disponível em: <<https://en.unesco.org/covid19/educationresponse>>. Acesso em: 27 set. 2021.
- [3] Resumo Executivo - Pesquisa Tic Domicílios. Cetic. Disponível em: <[https://cetic.br/media/docs/publicacoes/2/20201123115919/resumo\\_executivo\\_tic\\_do\\_m\\_2019.pdf](https://cetic.br/media/docs/publicacoes/2/20201123115919/resumo_executivo_tic_do_m_2019.pdf)>. Acesso em: 27 set. 2021
- [4] Ministério da Educação. Resolução nº 7, de 18 de dezembro de 2018. Disponível em: <[http://portal.mec.gov.br/index.php?option=com\\_docman&view=download&alias=104\\_251-rces007-18&category\\_slug=dezembro-2018-pdf&Itemid=30192](http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=104_251-rces007-18&category_slug=dezembro-2018-pdf&Itemid=30192)> Acesso em: 27 set. 2021.
- [5] MIT OpenCourseWare. Disponível em: <https://ocw.mit.edu/index.htm>. Acesso em: 1 out. 2021.

- [6] Abelson, Hal. (2008). The creation of OpenCourseWare at MIT. Journal of Science Education and Technology. 17. 10.1007/s10956-007-9060-8. Disponível em: <https://dspace.mit.edu/bitstream/handle/1721.1/37585/ocw-creation-preprint.pdf>. Acesso em: 25 set 2021.
- [7] TANEJA, Shilpi; GOEL, Anita. MOOC Providers and their Strategies. **IJCSMC**, Vol. 3, Issue. 5, May 2014, pg.222 – 228. Disponível em: [https://www.academia.edu/7054738/MOOC\\_Providers\\_and\\_their\\_Strategies](https://www.academia.edu/7054738/MOOC_Providers_and_their_Strategies) . Acesso em: 1 out 2021.
- [8] CORNEJO-VELAZQUEZ, Eduardo & Clavel-Maqueda, Mireya & Perez, Hector & Lyubimova, Ekaterina. (2020). Business Model of Learning Platforms in Sharing Economy. Electronic Journal of e-Learning. 18. 102-113. 10.34190/EJEL.20.18.1.008. Disponível em: <<https://files.eric.ed.gov/fulltext/EJ1245290.pdf>>. Acesso em: 25 set 2021.
- [9] RIBEIRO, L. O. M.; CATAPAN, A. H. PLATAFORMAS MOOC E REDES DE COOPERAÇÃO NA EAD. EmRede - Revista de Educação a Distância, v. 5, n. 1, p. 45-62, 16 mar. 2018.
- [10] Figma. Disponível em: <https://www.figma.com/>. Acesso em: 1 out. 2021
- [11] FREEMAN, Eric; ROBSON, Elizabeth. Use sua cabeça! HTML e CSS. [s.l]:Alta Books, 1 junho 2015.
- [12] MORRISON, Michael. Use sua Cabeça! Javascript. [s.l]:Alta Books, 4 setembro 2020.
- [13] TATROE, Kevin; MACINTYRE, Peter; LERDORF, Rasmus. Programming PHP. "O'Reilly Media, Inc.", v. 1, f. 270, 2013. 540 p.
- [14] SCHWARTZ, Baron; ZAITSEV, Peter; TKACHENKO, Vadim. High Performance MySQL: Optimization, Backups, and Replication. "O'Reilly Media, Inc.", v. 3, f. 413, 2012. 826 p.
- [15] STAUFFER, Matt. **Laravel: Up & Running**: A Framework for Building Modern PHP Apps. 2 ed. "O'Reilly Media, Inc.", 2019. 690 p.
- [16] LOPEZ, Lionel. **Vue**: Step-by-step Guide to Mastering Vue.js from Beginner to Advanced. 2017. 146 p.
- [17] NIELSEN, J., and MOLICH, R. (1990). Heuristic evaluation of user interfaces, Proc. ACM CHI'90 Conf. (Seattle, WA, 1–5 April), p. 249–256. Disponível em: <http://cs.ashoka.edu.in/cs102/papers/heuristic-evaluation-of-user-interfaces-nielsen.pdf>.
- [18] BALSAMIQ. Disponível em: <https://balsamiq.com/>. Acesso em: 1 out. 2021.

- [19] JOBLOVE, George H.; Greenberg, Donald. COLOR SPACES FOR COMPUTER GRAPHICS. in SIGGRAPH '78: Proceedings of the 5th annual conference on Computer graphics and interactive techniques, 5., 1978, Atlanta, GA, USA. New York, NY, USA : Association for Computing Machinery, 1978. p. 20-25. Disponível em: <https://doi.org/10.1145/965139.807362>. Acesso em: 29 set. 2021
- [20] Miro. Disponível em: <https://miro.com/>. Acesso em: 1 out. 2021
- [21] CLARKE-SALT, Justin. SQL Injection Attacks and Defense, 2nd Edition, f. 288. 2008. 576 p.
- [22] Security | Vue. Disponível em: <https://v3.vuejs.org/guide/security.html>. Acesso em: 29 set. 2021
- [23] PROVOS, Niels; MAZIERES, DAVID. A Future-Adaptable Password Scheme. in: Proceedings of the FREENIX Track: 1999 USENIX Annual Technical Conference, 10., 1999, Monterey, Califórnia, EUA. junho, 1999. Disponível em: <<https://www.usenix.org/legacy/events/usenix99/provos/provos.pdf>>. Acesso em: 29 set. 2021.
- [24] STEVENS, Hallam. Hans Peter Luhn and the birth of the hashing algorithm. IEEE spectrum, v. 55, n. 2, p. 44-49, fevereiro, 2018. Disponível em: <[ieeexplore.ieee.org/document/8278136](https://ieeexplore.ieee.org/document/8278136)>. Acesso em: 29 set. 2021.