

Plataforma para organização sistemática de material didático *online*

Leandro Felipe Castilhos¹, Yan Neto dos Santos¹, Jose Dario Pintor da Silva¹

¹Curso de Ciência da Computação – Instituto Federal de Santa Catarina(IFSC)
Rua Heitor Vilas Boas, 222 - São Francisco, Lages - SC, 88506-400

{leandro.fc, yan.n}@aluno.ifsc.edu.br

{jose.silva}@ifsc.edu.br

Abstract. *Teaching using digital media has been increasingly used as a facilitator of teaching and learning. With the growth of this educational object of teaching, platforms have emerged that help in the teaching-learning process. This article presents the creation of an online platform for the systematic organization of teaching material focused on a better user experience. The system allows a teacher to make materials organized in modules and classes available in the form of a course for interested students. From an evaluation carried out, the system proved to be attractive and intuitive.*

Resumo. *O ensino com uso de mídias digitais vem sendo cada vez mais utilizado como facilitador do ensino e aprendizagem. Com o crescimento desse objeto educacional de ensino surgiram plataformas que auxiliam no processo de ensino aprendizagem. Este artigo apresenta a criação de uma plataforma online para organização sistemática de material didático focada em uma melhor experiência de usuário. O sistema permite um professor disponibilizar materiais organizados em módulos e aulas no formato de um curso para alunos interessados. A partir de uma avaliação realizada o sistema mostrou-se atrativo e intuitivo.*

1. Introdução

O avanço da tecnologia da informação trouxe diversas mudanças na vida das pessoas. Cada vez mais pessoas estão utilizando dispositivos eletrônicos, com acesso à internet, que facilitam a troca de informações entre pessoas e instituições. A internet está presente em vários contextos do nosso cotidiano, seja para mandar um *e-mail*, acessar redes sociais, usar ferramentas empresariais, buscar por informações ou notícias e até mesmo realizar compras nas diversas lojas virtuais existentes. Assim, muitas áreas precisaram evoluir e se adaptar para tirar proveito dessa nova realidade tecnológica digital.

O Programa Internacional de Avaliação de Aluno (PISA - Programme for International Student Assessment) de 2018 revela que os estudantes brasileiros estão dois anos e meio abaixo dos países da Organização para a Cooperação de Desenvolvimento Econômico (OCDE) em relação ao nível de escolarização de proficiência em leitura. Em ciências o resultado do exame revela que 55% dos estudantes não atingiram o nível básico, já em matemática 68,8% dos alunos estão no pior nível de proficiência. Assim, se faz

necessário lançar mão de abordagens que articulem a tecnologia da informação com estratégias pedagógicas de ensino e aprendizagem para que possam auxiliar na superação desse problema.

O ensino a distância é uma realidade há muito tempo no Brasil, e vem crescendo no ensino, sendo uma opção para ampliar o acesso dos alunos a aprendizagem. Com essa modalidade de ensino surgiram variadas ferramentas e plataformas que servem para organizar essa maneira diferente da tradicional de estudar, um exemplo seria a plataforma *Udemy*, que possui cursos de diversas áreas, sendo alguns gratuitos e outros pagos. Outra plataforma é o *Moodle*, que também disponibiliza cursos *online* como apoio para instituições de ensino, como o Instituto Federal de Santa Catarina (IFSC) e a Universidade Federal de Santa Catarina (UFSC).

Essas plataformas auxiliam no processo de ensino e aprendizagem *online*, porém suas funcionalidades são customizadas de forma genérica para atender um número maior de pessoas. O desenvolvimento genérico acaba por impactar a experiência do usuário, pois as funcionalidades não são personalizadas para atender as especificidades de uma comunidade escolar local. Surgiu, então, a ideia de implementar uma plataforma com características de usabilidade mais práticas, voltadas a uma comunidade menor, um câmpus de uma Instituição, por exemplo.

Portanto, o objetivo do projeto é desenvolver uma ferramenta para auxiliar na organização sistemática do material acessado através da internet, que possibilite fácil navegação para o usuário personalizado para atender as necessidades da comunidade de ensino local. Para alcançar esse resultado foram definidos os seguintes objetivos específicos:

- Realizar estudos sobre plataformas educacionais e usá-las como referência;
- Estudar técnicas de usabilidade;
- Implementar uma plataforma educacional, para realizar a organização sistemática de material didático *online*;
- Avaliar os módulos desenvolvidos.

Em relação às metodologias de pesquisa este trabalho se classifica de acordo com a natureza como pesquisa aplicada, pois o objetivo é gerar conhecimento na prática e solucionar uma questão da nossa realidade. No ponto de vista da abordagem do problema é qualitativa, pois não foram utilizados instrumentos estatísticos na análise dos dados. Conforme Zanella (2013), a pesquisa exploratória tem a finalidade de ampliar o conhecimento a respeito de algo específico, portanto relacionado aos objetivos a pesquisa é exploratória. Referente ao ponto de vista dos procedimentos técnicos será uma pesquisa bibliográfica, baseada em artigos disponibilizados recentemente e que são similares aos nossos objetivos.

O projeto será iniciado com um estudo sobre plataformas educacionais já existentes e a partir delas usá-las como referência para o projeto. Os artigos para estudos serão pesquisados em bibliotecas digitais que disponibilizam conteúdos relevantes de periódicos e eventos.

Na sequência também será realizado um estudo sobre técnicas de usabilidade, mais especificamente *User Experience* (UX), com o intuito de melhorar a plataforma e deixá-la ainda mais acessível para os alunos e tutores. Esse estudo será realizado através

de conteúdos sobre o assunto encontrados *online*, em livros e também em artigos.

A implementação da plataforma se iniciou com a escolha das ferramentas e linguagens que foram utilizadas, além de um gerenciamento que melhor se adeque ao projeto e ao tempo disponível para o desenvolvimento. O gerenciamento escolhido foi a estrutura *scrum*, para uma implementação mais organizada e para que todos os integrantes ficassem a par da fase onde se encontrava o projeto. O desenvolvimento se iniciará na prática com a criação do banco de dados com uma estrutura inicial que pode ser suplementado conforme a necessidade da plataforma. Em sequência tem-se o objetivo de iniciar a programação da ferramenta por partes definidas com o gerenciamento *scrum*.

Por fim, realizou-se uma avaliação da plataforma em geral, pelos desenvolvedores e também por usuários externos para obter uma visão diferenciada do sistema com o intuito de encontrar falhas e pontos que podem ser melhorados. Esta avaliação ocorrerá através de testes e do envio de um questionário *online* para auxiliar na coleta dos *feedbacks*.

Além desta seção introdutória, este documento é composto por outras três seções. A seção dois apresenta os assuntos relacionados ao entendimento do trabalho e também trabalhos similares. A seção três, registra o desenvolvimento do trabalho. E, na seção quatro, as conclusões sobre o trabalho são apresentadas.

2. Referencial Teórico

Esta seção compõe-se dos temas que servem para auxiliar no desenvolvimento do projeto. A seção foi estruturada da seguinte forma: Educação a distância e Ferramentas digitais, Experiência do usuário, Trabalhos similares e Tecnologias utilizadas.

2.1. Educação a Distância e Ferramentas Digitais

O ensino por outros meios além do presencial existe há muito tempo, mesmo antes de a tecnologia estar presente em nosso cotidiano. Anteriormente, o ensino não presencial era realizado de outras maneiras, como a utilização de correspondências. Alves (2001) cita que os primeiros dados conhecidos sobre educação a distância são do século XX, e que somente na década de 1990, é que a maior parte das Instituições de Ensino Superior brasileiras mobilizou-se para a Educação a Distância, com o uso de novas tecnologias de informação e comunicação.

No ano de 2020, com a pandemia do Covid-19, ficou ainda mais evidente a utilização de ferramentas digitais para auxiliar o ensino não presencial, com a publicação da portaria nº 343, de 17 de Março de 2020, que dispõe sobre a substituição das aulas presenciais por aulas em meios digitais enquanto durar a situação de pandemia. O ensino não presencial não se caracteriza totalmente como EaD, porém ferramentas utilizadas para EaD, também puderam ser utilizadas nesse método de ensino.

A educação não presencial é um método de ensino fora da sala de aula, onde é feita a utilização, na maioria das vezes, de meios digitais para realizar a comunicação. Para Hack (2011) a Educação a Distância (EaD) é vista como uma modalidade de realizar o processo de aprendizagem no momento em que o encontro presencial do educador e do aluno não acontece. Portanto, professores e alunos ficam separados fisicamente, mas podem estar conectados através de múltiplas tecnologias, como a internet (Moran, 2000).

Ainda segundo Moran (2000) a educação a distância pode ser feita nos mesmos níveis que o ensino regular. Porém, é mais adequado para a educação de adultos, principalmente para aqueles que já têm experiência na aprendizagem individual e de pesquisa, como acontece no ensino de graduação e pós-graduação.

No ensino a distância, mesmo o modo de ensinar sendo a distância o educador tem um papel imprescindível, pois ele coopera com o aluno ao formular problemas, provocar interrogações ou incentivar a formação de grupos (Hack, 2011).

Para fazer a comunicação entre aluno e docentes que acontece de forma virtual, é feita a utilização de ferramentas e plataformas que auxiliam na organização sistemática do material didático. Para Schlemmer e Fagundes (2000), Ambientes Virtuais de Aprendizagem, Sistemas Gerenciadores de Educação a Distância, *Software* de Aprendizagem Colaborativa são algumas denominações para os *softwares* desenvolvidos especificamente para o gerenciamento da aprendizagem via *Web*. São sistemas que fazem as funções dos softwares nos processos de comunicação e integração mediadas por computadores nas metodologias dos cursos online (Schlemmer e Fagundes, 2000).

Existem diversos ambientes nesse sentido, um deles é a *Udemy*. *Udemy* é uma plataforma de cursos *online*, que possui uma biblioteca com 115 mil cursos, e 40 milhões de alunos até o fim do primeiro semestre do ano de 2021¹. Os cursos são de variadas categorias, onde são criados, administrados e de propriedade do instrutor. A base dos cursos são as aulas, que podem incluir vídeos, apresentações e textos, além do tutor poder adicionar recursos práticos para melhorar o aprendizado dos alunos.

Alura é um ambiente brasileiro que disponibiliza cursos *online* para estudar variados temas relacionados a tecnologia, o acesso funciona a partir de planos, diferente da *Udemy*, onde os cursos podem ser comercializados individualmente, porém a base para os cursos são similares, podemos encontrar vídeos, *slides*, textos e também atividades práticas.

Modular Object-Oriented Dynamic Learning Environment, Ambiente de Aprendizado Modular Orientado ao Objeto ou apenas Moodle é um ambiente virtual de aprendizado utilizado com apoio ao ensino a distância por alunos e professores. Funciona como uma sala de aula *online*, os professores conseguem disponibilizar conteúdos didático, propor tarefas e discussões para o aluno. Por ser um *software* de código aberto disponível para qualquer instituição, o *Moodle* é utilizado em diversos países. No Brasil sua presença está em sua maior parte nas universidades como na Universidade de São Paulo (USP), Universidade Federal da Bahia (UFBA), Instituto Federal de Santa Catarina (IFSC), entre outras.

O Moodle é definido como *Massive Open Online Course* (MOOC), que é um termo para os cursos abertos *online* em massa, cursos que são definidos por suas características de serem abertos e gratuitos, realizados de forma não presencial e não possuir uma quantidade predefinida de participantes, ou seja, disponibilizados para milhares de pessoas (Garrido et al., 2018b).

Além dessas plataformas de curso, existem outras que auxiliam no ensino não presencial. Portanto, a expansão da tecnologia da informação e comunicação tem imple-

¹<https://www.udemy.com>

mentado meios para a construção do conhecimento, tornando, a aprendizagem mais rica, motivadora e significativa (Araújo, 2013).

2.2. Experiência do Usuário

Para os sistemas serem melhores avaliados é utilizado a Experiência de Usuário, em inglês, *User experience* ou apenas UX, termos que foram popularizados na década de 90 por Donald Norman.

Os sistemas computacionais interativos são avaliados para que haja melhorias na qualidade de seu uso, como aumento da produtividade do usuário, redução do risco de erros cometidos, redução dos custos de treinamento e suporte, entre outros quesitos (Gomes et al., 2015). Um dos grandes desafios, é poder unir o *design* institucional com a experiência de usuário, com uma interferência mínima de um aspecto no outro.

A experiência de usuário é um conjunto de elementos que influenciam na experiência e percepção que um usuário tem sobre um produto, que está ligada diretamente a interação homem-computador (IHC) sendo definida por Baranauskas e Rocha (2000) como disciplina preocupada com o *design*, avaliação e implementação de sistemas computacionais interativos para uso humano.

A experiência do usuário com um produto é afetada diretamente pela forma como este é planejado e concebido, isto é, o seu *design*. O conceito de *design* está relacionado ao projeto, sendo empregado no sentido de desenho de um esquema, planejamento e concepção de um artefato ou serviço, com o objetivo de resolver problemas ou demandas específicas. (Grilo, 2019)

Como ainda cita Grilo (2019), o entendimento de que *Design* é projetar para experiência é explicado também pela impossibilidade de profissionais determinarem ou controlarem fatores externos às configurações do produto que afetam a experiência com este, como a circunstância de uso e o próprio usuário do artefato.

Aarron Walter, que atuou como diretor de experiência do usuário na *Mailchimp* e compõe o quadro executivo na *InVision*, propõe uma articulação entre usabilidade e experiência do usuário. Segundo Walter (2011), um produto deve oferecer as seguintes qualidades interdependentes: ser funcional, confiável, usável e agradável. (Grilo, 2019)

Já o *design* institucional é a ciência da criação de currículo de instrução que se destina a produzir resultados de aprendizagem específicos, baseados não apenas na pesquisa pedagógica, mas também nas práticas de instrução contemporâneas. (Garrido et al., 2018a)

Ademais, no caso de sistemas para o ensino a distância com *designs* institucionais, que de acordo com Garrido et al. (2019), representa 84% dos sistemas MOOCs, nota-se em sua maioria, uma falta de cuidado com a experiência de usuário, o que acaba por influenciar consideravelmente no processo de ensino-aprendizagem, visto que a falta de compreensão do sistema afeta no tempo e na qualidade de ensino, afetando por fim a correta aplicação de um *design* institucional.

2.3. Trabalhos Similares

Nesta subseção são apresentados trabalhos relacionados ou com semelhança ao tema proposto neste trabalho.

Foi realizado uma pesquisa através da biblioteca digital SBC *OpenLib*², onde foram seguidos alguns passos para buscar artigos relevantes ao tema desta pesquisa e que possam auxiliar no desenvolvimento do trabalho.

Morais et al. (2020) apresenta uma proposta de desenvolvimento de uma ferramenta para auxiliar na educação de jovens e adultos (EJA), denominada EJA Conectada. Ferramenta com o intuito de ser usada em sala de aula para contribuir no processo de alfabetização. Na ferramenta são apresentados diversas categorias de conteúdo, separados em módulos.

Para a modelagem da aplicação foi utilizado o método de *Design* de Interação, que se caracteriza em realizar um levantamento de requisitos, criar alternativas de *design*, prototipar e avaliar (Rogers et al., 2013). Para o desenvolvimento foram utilizadas algumas ferramentas, como o *GitHub* para armazenamento, *Sublime Text Editor*, como editor de texto. A linguagem escolhida para a implementação foi C#, além de *JavaScript*, *HTML5* e *CSS3* para o desenvolvimento da interface.

A ferramenta ficou dividida em quatro módulos, que avançam conforme a evolução do aluno, como exemplo, o módulo 1 se trata da apresentação das letras e o módulo 4 trabalha com a produção de textos.

Como uma ferramenta *online*, o EJA Conectada pode ser facilmente acessado através de um computador conectado a internet. A plataforma pode ser utilizada pelos professores em sala de aula para possibilitar a exploração de outras práticas além das do sistema, complementando e fazendo parte do conteúdo didático (Morais et al., 2020).

Foi realizada também a análise do artigo de Rhis (2018) sobre o Desenvolvimento de uma plataforma educacional para disseminação do conhecimento. No trabalho o autor observou que os docentes locais utilizavam de ferramentas que não atendiam todas as demandas educacionais, comprometendo o desenvolvimento de um trabalho melhor. Surgiu então a necessidade de desenvolver uma plataforma com o intuito de suprir essas carências do meio acadêmico. A plataforma foi chamada CEVIBRA.

O desenvolvimento da plataforma iniciou com uma análise de requisitos, realizada junto com os técnicos e professores da UFVJM. Após esse levantamento de dados foi iniciado a implementação do banco de dados na ferramenta *MYSQL Workbench*. Somente depois de o banco de dados estar pronto foi feita a escolha da linguagem de programação a ser utilizada, a que melhor atendeu a plataforma foram PHP e *JavaScript* com estrutura *Model-View-Controller* (MVC). O MVC consiste na separação da aplicação em três camadas, que seriam as operações relacionadas ao aplicativo, a exibição do aplicativo e por último a interação do usuário com as duas outras camadas (Krasner e Pope, 1988).

O CEVIBRA, além de ofertar cursos *online*, também tem objetivo de ser algo a mais que uma plataforma de cursos, portanto, nele também é possível gerenciar eventos, validar certificados, obter informativos acadêmicos e ter acesso a serviços de *marketing*.

O trabalho de Renz et al. (2019) mostra o desenvolvimento de um *software* para complementar o processo de ensino e aprendizado em contabilidade. O objetivo de construir o programa surgiu após ser verificado que a maioria das profissionais na área fazem a utilização de *softwares* para auxiliar no trabalho.

²<https://sol.sbc.org.br/index.php/indice>

O programa foi desenvolvido por alunos bolsistas voluntários utilizando da linguagem *JavaScript*, no qual foi possível realizar a implementação das funcionalidades relevantes e funções de cálculos e também a adição dos conteúdos teóricos, tornando o sistema para algo além de só atividades práticas, mas também para lecionar um dado assunto ou conteúdo. A ferramenta está dividida em quatro partes, a página inicial realiza a apresentação do projeto, a segunda página é utilizada para mostrar informações sobre a equipe desenvolvedora, na terceira é trabalhado o conteúdo teórico pertinente a contabilidade, e a última página que pelos desenvolvedores é considerada a principal, em que se trabalha com a prática de exercícios, nela é possível o aluno realizar 14 atividades já predefinidas que estão divididas em três níveis de dificuldade e também pode se realizar uma atividade livre proposta pelo próprio professor.

Os testes do *software* foram realizados no curso técnico de administração, onde foi realizado a comparação entre duas turmas, uma sem a utilização da ferramenta e outra fazendo o uso da mesma. A turma experimental do programa obteve um melhor desempenho na questão de notas, assim mostrando um êxito ao projeto.

2.4. Tecnologias Utilizadas

Para o desenvolvimento da plataforma, foi utilizado principalmente a linguagem C# com o *framework ASP.NET Core* que auxilia no desenvolvimento de aplicações web com a IDE *Visual Studio*. A escolha se deu principalmente pela experiência da equipe com a utilização do *framework*, o mesmo também é bem documentado e pode ser conferido no site da própria *Microsoft*³.

O ASP.NET Core também é constantemente atualizado e mantido pela *Microsoft* o que facilita a segurança e o desenvolvimento usando novas tecnologias e/ou conceitos na programação.

Fora a linguagem C# também utilizou-se as linguagens *JavaScript*, HTML5 e CSS5, incluindo o *framework Bootstrap* que faz o uso destas linguagens para o desenho e *design* das telas no geral melhorando a experiência de usuário, gerenciando principalmente o lado do cliente na plataforma. Todas as linguagens, são regidas e atualizadas pela organização *World Wide Web (W3C)*⁴.

O Consórcio W3C é um consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a *Web*. Liderado pelo inventor da web Tim Berners-Lee e o CEO Jeffrey Jaffe, o W3C tem como missão Conduzir a *World Wide Web* para atingir todo seu potencial, desenvolvendo protocolos e diretrizes que garantam seu crescimento de longo prazo.

O banco de dados escolhido foi o *MySQL Community* que já é bastante conceituado, é gratuito e *open source*, facilitando a busca pela resolução de problemas que possam vir a acontecer. Atualmente sua versão estável é a 8.0.26 sendo mantido pela *Oracle* e documentado pela mesma⁵.

³<https://docs.microsoft.com/pt-br/aspnet/core/>

⁴<https://www.w3c.br>

⁵<https://dev.mysql.com/doc/refman/8.0/en/>

3. Desenvolvimento

Nesta seção encontra-se o desenvolvimento do sistema. Esta seção está subdividida em três subseções que descrevem o processo de desenvolvimento, sendo elas Metodologia de desenvolvimento, Ferramenta e Resultados.

3.1. Metodologia de desenvolvimento

O desenvolvimento deste trabalho iniciou com a busca de trabalhos relacionados e ferramentas educacionais para preenchimento do referencial teórico e obtenção de conhecimentos relacionados que serviram de auxílio para escrita deste artigo e implementação da ferramenta. A busca consistiu na procura de artigos relacionados seguindo os passos sugeridos por Wazlawick (2009). O primeiro passo foi listar títulos de periódicos e eventos que possam ter artigos relacionados, como o Simpósio Brasileiro de Informática na Educação (SBIE), em seguida foi efetuado uma nova busca para selecionar artigos publicados nos últimos três anos nos veículos utilizando de palavras chaves como “ferramenta de ensino” e “plataforma de ensino”. Após se obter essa lista, foi realizada a leitura do resumo dos artigos selecionados, e foi possível realizar a classificação dos mesmos por relevância, sendo “alta”, “média”, ou “baixa”. A leitura ampla, foi iniciada nos artigos com relevância alta e se necessário realizar a leitura dos demais, com o intuito de conseguir conhecimento dos principais conceitos e ideias relacionados ao tema e também acrescentar novos artigos citados pelos autores. A partir daí foi realizado a análise e o preenchimento do documento de requisitos ⁶ de modo a levantar as funcionalidades a serem atendidas pela plataforma. No documento foi descrito as funções, interfaces e o comportamento esperado do sistema.

Realizaram-se também estudos sobre a usabilidade de modo a definir quais técnicas se encaixariam melhor no projeto para melhora da interação humano-computador. Através dos estudos foi notado que a harmonização das cores influenciam diretamente no desenvolvimento de uma boa experiência de usuário, decidiu-se então selecionar uma paleta de cores para uso na aplicação *web*. Com a paleta de cores já definida foi aplicado a regra 60-30-10, onde a cor primária deve ser usada em 60% do site, a cor secundária deve ser em 30%, e a cor de destaque deve ser em apenas 10% (Ferris e Zhang, 2016). Portanto, a cor primária é utilizada em boa parte do site, um exemplo seria no *background*, os 30% são utilizados para dar apoio a cor principal, pode ser usado em menus, por exemplo. A cor destaque que ocupa apenas 10% da aplicação é usada para destacar algo, como o próprio nome sugere, utilizada em botões, alerta, entre outros.

Para gerenciamento do processo desenvolvimento foi utilizado a metodologia *Scrum*. *Scrum* é uma estrutura para gerenciamento de projetos, que permite o desenvolvimento de formar mais ágil. Segundo Sutherland (2019), essa metodologia funciona com a definição de objetivos sequenciais que devem ser concluídos em um período definido. No *scrum* toda a equipe consegue acompanhar o que está sendo desenvolvido e o que ainda precisa ser desenvolvido, assim possibilitando verificar em qual fase o projeto se encontra. Os requisitos criados anteriormente foram adicionados ao *Product Backlog*, para serem desenvolvidos conforme decidido na *Sprint Planning Meeting*. As *Sprint Planning Meeting* são reuniões realizadas antes de cada *Sprint*. As *Sprints* são os períodos de desenvolvimento, no caso deste projeto, foi decidido que cada *Sprint* teria uma duração de duas

⁶<https://bityli.com/jlWk06>

semanas.

Os padrões de projeto no contexto de projetos de *software* são soluções para problemas comuns, adaptados para dentro do projeto que está em desenvolvimento para solucionar um problema. São descrições de objetos e classes comunicantes que precisam ser personalizadas para resolver um problema geral de projeto num contexto particular (Gamma et al., 2000).

Inicialmente, pensou-se em desenvolver toda a plataforma utilizando o MVC, porém foram implementados alguns outros padrões de projeto para facilitar o desenvolvimento e a manutenção, sendo eles: *Façade*, *Factory Method* e *Services*.

E no contexto desse trabalho esses três padrões se adaptaram muito bem, pois utilizando o *Façade* e o *Services* isola-se a responsabilidade de cada classe no código, sendo assim, o fluxo no código ficou da seguinte forma como mostra a Figura 1.

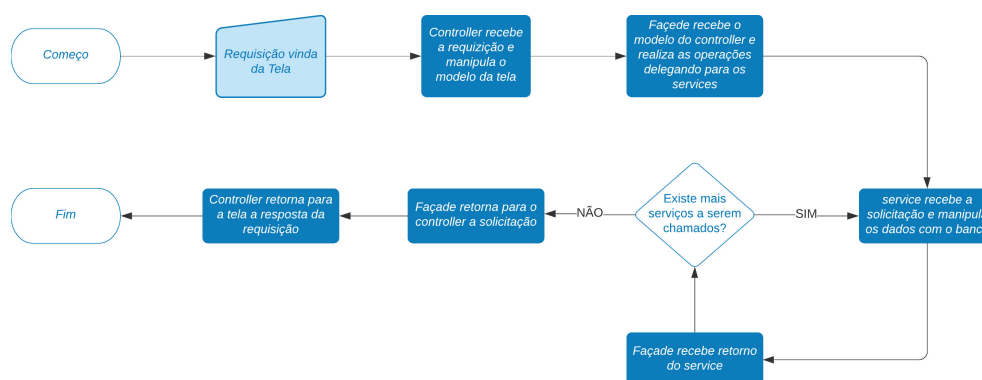


Figura 1. Fluxograma de uma requisição

Como se vê na Figura 1, foi isolado a responsabilidade de manipular o modelo da tela apenas com o *Controller*, a responsabilidade de delegar as ações dos diversos possíveis modelos é do *Façade*, por fim, realizar as ações é uma responsabilidade dos *Services*. Aqui cada padrão assume a forma de uma classe, podendo ter diversas classes *Façade* e *Services* na estrutura do código. Na Figura 2 temos um exemplo em código do padrão *Façade*.

Já para o padrão *Factory Method* sua responsabilidade consiste em um passo antes do fluxograma, no ato do *login*, por isso não está descrito no fluxograma da Figura 1. Ele foi útil para construir um objeto para o controle de usuários e autenticações, reutilizando a biblioteca *Identity* do ASP.NET Core. Falou-se mais sobre ela a frente neste artigo.

Após toda a implementação, foi realizada uma avaliação da plataforma em geral, incluindo testes dos casos de uso e das funcionalidades. Estes testes foram realizados por professores da instituição, após os testes e análise dos resultados foi possível realizar alterações para fins de melhorar a experiência de usuário.

3.2. Ferramenta

Nesta subseção foram apresentadas informações sobre o desenvolvimento da ferramenta e sua estrutura. A subseção estará dividida em outras subseções sendo elas Banco de dados, Telas e *Layouts*, *Chat*, *Login* e Registro, Módulos e os módulos presentes na ferramenta:

```

1 namespace TutorEAD.app.Facades.Implements
2 {
3     public class CourseFacade : ICourseFacade
4     {
5         private readonly CoursesService _coursesService;
6         private readonly ModulesService _modulesService;
7         private readonly ClassService _classService;
8         private readonly ClassContentsService _classContentsService;
9         private readonly UserCoursesService _userCoursesService;
10        public CourseFacade(CoursesService coursesService, ModulesService modulesService, ClassService classService,
11            ClassContentsService classContentsService, UserCoursesService userCoursesService)
12        {
13            _coursesService = coursesService;
14            _modulesService = modulesService;
15            _classService = classService;
16            _classContentsService = classContentsService;
17            _userCoursesService = userCoursesService;
18        }
19        public List<CourseModel> GetAllAvailableCourses() => _coursesService.GetAllAvailable().ToList();
20        public List<CourseModel> GetCourseByUserId(String userId) => _userCoursesService.GetCoursesByUserId(userId).ToList();
21        public CourseModel GetCourseById(Int64 id) => _coursesService.GetById(id);
22        public CourseModel GetFullCourseById(Int64 id) => _coursesService.GetFullById(id);
23
24        public async Task<ActionResult> InsertCourse(CourseModel model)
25        {
26            if (String.IsNullOrWhiteSpace(model.Name)) return new BadRequestObjectResult("O nome deve ser preenchido.");
27            model.CreatedDate = DateTime.Now;
28            return await _coursesService.Insert(model);
29        }
30    }
31 }

```

Figura 2. Exemplo de código-fonte do padrão *Façade*

Módulo Tutor, Módulo Aluno e Módulo Administrador. As subseções dos módulos apresenta o funcionamento da ferramenta, considerando a visão de cada ator existente na plataforma, e mostra quais as funcionalidades disponíveis.

3.2.1. Banco de dados

O banco escolhido para a plataforma, foi o MySQL. Sendo assim iniciou-se uma modelagem inicial do banco com as principais tabelas que compõe o sistema, desde usuários, a cursos e tarefas. O diagrama inicial do banco pode ser observado na Figura 3.

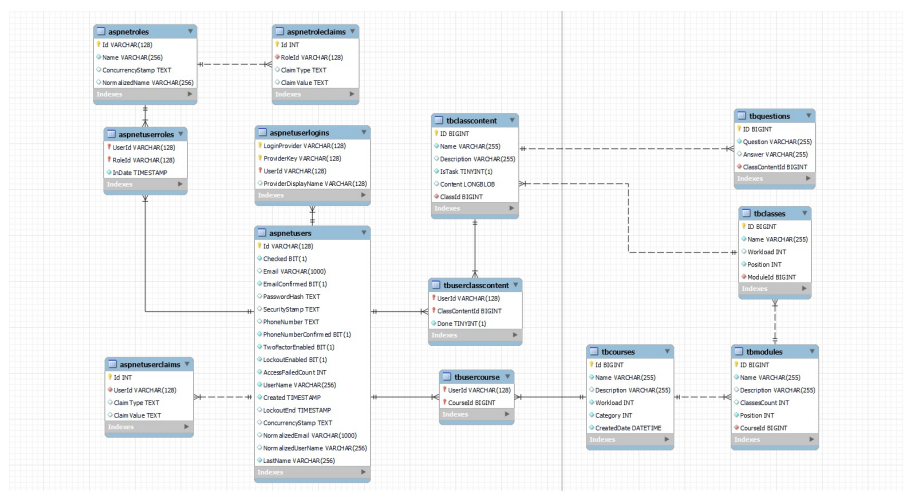


Figura 3. Diagrama Inicial do banco de dados

Utilizando da biblioteca do *Entity Framework Core (EFCore)*⁷ o acesso e a

⁷<https://docs.microsoft.com/pt-br/ef/core/>

manipulação dos dados ficou extremamente versátil e claro, sem *queries* longas do modelo *Data Access Object*(DAO), por exemplo.

O EFCore necessita apenas de um Modelo em C# equivalente à tabela em banco, então basta ser criado o modelo e relacioná-lo aquela tabela, depois disso, todas as operações de *Create, Read, Update, Delete* (CRUD) podem ser feitas através do próprio modelo e foram realizadas pelas classes *Services*, que é um padrão de projeto já próprio para essa ação.

Na Figura 4 temos um exemplo do padrão de projeto *Services*, com o método *Insert*:

```
1 namespace TutorEAD.app.Services
2 {
3     public class CoursesService
4     {
5         private readonly ApplicationDbContext _dbContext;
6         public CoursesService(ApplicationDbContext dbContext)
7         {
8             dbContext.ChangeTracker.AutoDetectChangesEnabled = false;
9             _dbContext = dbContext;
10        }
11        public IQueryable<CourseModel> GetAllAvailable() => _dbContext.Courses;
12        public CourseModel GetById(Int64 id) => _dbContext.Courses.FirstOrDefault(x => x.Id == id);
13        public CourseModel GetFullById(Int64 id) => _dbContext.Courses
14            .Include(x => x.Modules)
15            .ThenInclude(x => x.Classes)
16            .ThenInclude(x => x.ClassContents)
17            .ThenInclude(x => x.Tasks)
18            .ThenInclude(x => x.Forms)
19            .ThenInclude(x => x.FormsQuestions)
20            .ThenInclude(x => x.Question)
21            .FirstOrDefault(x => x.Id == id);
22        public async Task<ActionResult> Insert(CourseModel model)
23        {
24            try
25            {
26                _dbContext.Courses.Add(model);
27                await _dbContext.SaveChangesAsync();
28                return new CreatedResult("Courses", model);
29            }
30            catch (Exception ex)
31            {
32                return new BadRequestObjectResult(ex.Message);
33            }
34        }
35    }
}
```

Figura 4. Exemplo de código-fonte do padrão *Services*

3.2.2. Telas e Layouts

O *layout* geral da ferramenta foi pensando para promover uma melhor experiência do usuário, em maior parte foi utilizado do *framework* Bootstrap para construção dos componentes presentes na interface. Bootstrap é um *framework* para desenvolvimento de interface e *front-end* usando de HTML, CSS e JavaScript que são as linguagens escolhidas para o desenvolvimento das telas.

O desenvolvimento da interface da plataforma seguiu o padrão MVC, foram criados três *layouts* com os componentes que se mantêm ao trocar de uma página para outra, um *layout* para cada ambiente existente na plataforma. O primeiro sendo a área de *login* e registro, uma tela mais básica em sentido visual, contendo apenas um quadro focado totalmente para o *login* e o registro dos usuários, como mostra a Figura 5. O segundo

layout a ser acessado é o de *home*, é onde acontece toda a navegação da aplicação. Por último, a tela do curso, é composta por uma barra lateral e um quadro onde é gerado o conteúdo selecionado, se tornando diferente do *layout* inicial da ferramenta.

O *home* apresenta duas barras de menu, sendo uma no topo, a *navbar* e outra na lateral esquerda, a *sidebar*, sendo elas dois menus fixos facilitando a navegação pela plataforma. A *navbar* mostra dados do usuário logado, e traz um menu em cascata com opções para gerenciar a conta e a *sidebar* contém todos os menus referente ao gerenciamento de cursos, como Cadastro de cursos, Lista dos cursos e Meus cursos. Além dos menus há um quadro para que o conteúdo solicitado seja gerado.

Para o gerenciamento dos conteúdos é utilizado o padrão MVC, é feito o uso de *controllers* para fazer a comunicação dos *layouts* com os conteúdos gerados a cada página. Os conteúdos são chamados de *views* no projeto, onde cada tela vai possuir uma *view* que é chamada a cada troca de página combinada com seu *layout*.

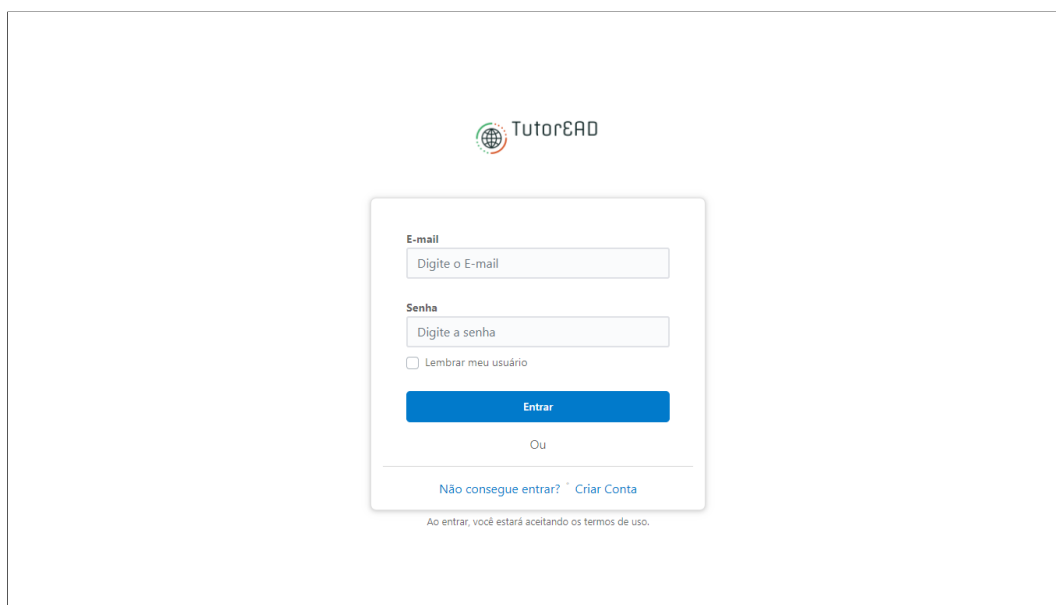


Figura 5. Tela de Login

3.2.3. Chat

Para a melhor interação entre os administradores da plataforma e os usuários foi incluído um *chat* de texto em tempo real na plataforma, para tal utilizou-se da biblioteca gratuita “JIVO”, que implementa em *javascript* todas as funcionalidades necessárias para a conversação entre os 2 pontos.

Na plataforma o *chat* fica disponível 100% do tempo em que o usuário estiver logado. Assim que uma mensagem é enviada da plataforma os administradores recepcionarão através do link: “<https://app.jivosite.com/chat/inbox>”.

Neste link já estarão disponíveis algumas informações para o administrador da plataforma, como o nome de usuário da plataforma e o endereço de *e-mail* do mesmo. Na Figura 6 é mostrado como é o chat na plataforma.

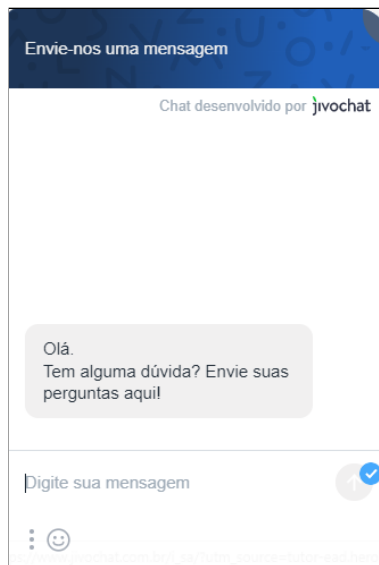


Figura 6. JivoChat

3.2.4. Login e Registro

A Plataforma funcionará com duas categorias de usuários principais, os Tutores/Professores e os Alunos, chamados estes de “Atores”. Cada Ator terá sua respectiva permissão de acesso.

Para gerenciar os acessos foi utilizado a biblioteca *Identity* do ASP.NET Core⁸. A biblioteca consiste em utilizar os pacotes das requisições para injetar informações do usuário logado através do que chamamos de *Claims*.

Os *Claims* são como a própria tradução sugere, as reivindicações do usuário, ou seja, neles estão contidas informações como o Nome do Usuário, o identificador único (ID) do mesmo, se ele é um ator do tipo aluno ou não, etc. Por padrão a biblioteca cria um *Claim* apenas com as informações básicas e é preciso no ato do *login*, subscrever o *Claim* criado pela biblioteca. O padrão de projeto *Factory Method* é utilizado para esse propósito. Na Figura 7 é mostrado um exemplo do padrão *Factory Method*:

O controle de acesso e de segurança é feito por um *token* criptografado salvo em *cookies* no ato do *login*, então, a requisição leva o *cookie* de autenticação e o servidor lê o mesmo e traduz as informações em um *Claim*, *token* tem um tempo máximo ocioso de uma hora. Extrapolado este tempo o servidor irá derrubar o usuário e o *login* deverá ser refeito.

3.2.5. Módulos

A modelagem dos módulos aconteceu através do documento de requisitos, onde com as informações preenchidas foi possível realizar a implementação dos casos de uso. Um caso de uso identifica os atores envolvidos em uma interação e dá nome a interação, acrescido

⁸<https://docs.microsoft.com/pt-br/aspnet/core/security/authentication/identity>

```

1 namespace TutorEAD.app.Factories
2 {
3     public class CustomClaimsPrincipalFactory :
4         UserClaimsPrincipalFactory<ApplicationUser>
5     {
6         protected HttpContext _httpContext
7         readonly protected ApplicationDbContext _dbContext;
8         public CustomClaimsPrincipalFactory(ApplicationDbContext userDbContext, UserManager<ApplicationUser> userManager,
9             IOptions<IdentityOptions> optionsAccessor, IHttpContextAccessor
10            httpContext) : base(userManager, optionsAccessor)
11     {
12         _httpContext = httpContext.HttpContext;
13         _dbContext = userDbContext;
14     }
15     override protected Task<ClaimsIdentity> GenerateClaimsAsync (ApplicationUser user)
16     {
17         var taskClaims = Task<ClaimsIdentity>.Run(async () => await base.GenerateClaimsAsync(user));
18         taskClaims.Result.AddClaim(new Claim("EmailConfirmed", user.EmailConfirmed.ToString()));
19         taskClaims.Result.AddClaim(new Claim(ClaimTypes.PrimarySid, user.Id.ToString()));
20         taskClaims.Result.AddClaim(new Claim("PhoneNumber", user.PhoneNumber ?? String.Empty));
21         taskClaims.Result.AddClaim(new Claim(ClaimTypes.Email, user.Email));
22         taskClaims.Result.AddClaim(new Claim(ClaimTypes.Expiration, DateTime.Now.Ticks.ToString()));
23         IList<String> roles = Task<ClaimsIdentity>.Run(() => GetRolesAsync(user)).Result;
24         taskClaims.Result.AddClaim(new Claim(ClaimTypes.Role, String.Join(',', roles)));
25         IList<Int64> coursesIds = Task<ClaimsIdentity>.Run(() => GetCoursesIdsAsync(user)).Result;
26         taskClaims.Result.AddClaim(new Claim(ClaimTypes.GroupSid, String.Join(',', coursesIds)));
27         return taskClaims;
28     }
29     private IList<String> GetRolesAsync(ApplicationUser user) => _dbContext.ApplicationUserRole.Where(x => x.UserId ==
30         user.Id).Select(x => x.Role.Name).ToList();
31     private IList<Int64> GetCoursesIdsAsync(ApplicationUser user) => _dbContext.UserCourses.Where(x => x.UserId ==
32         user.Id).Select(x => x.CourseId).ToList();
33 }
34 }

```

Figura 7. Exemplo de código-fonte do padrão *Factory Method*

por informações que descrevem a interação com o sistema (Sommerville, 2011). Atores nos casos de uso são os usuários que farão o uso da ferramenta, portanto os casos de uso representam a interação dos atores com as funcionalidades do sistema. No sistema há três atores já apresentados anteriormente e com os casos de uso é possível ter uma noção melhor de cada ator presente e quais são as funcionalidades pertencentes a cada um e as comuns a todos. Nas próximas subseções são apresentadas as funcionalidades e os principais casos de uso.

3.2.6. Módulo Tutor

O ator tutor na aplicação será como um professor. Ele disponibilizará cursos, fará correções e dará notas para os alunos, além das outras funcionalidades que estão disponíveis para todos os usuários, como visualizar os cursos listados. No decorrer desta subseção apresentasse detalhes das ações que o tutor pode realizar.

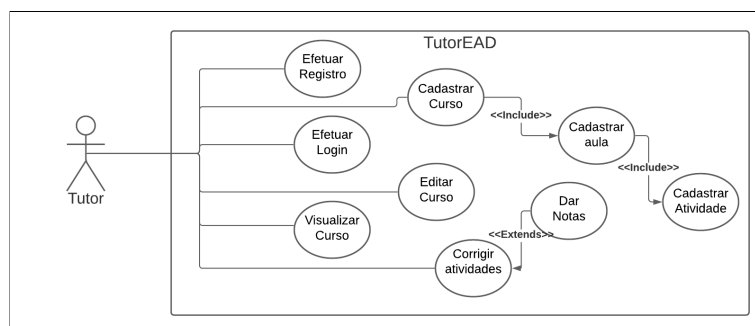


Figura 8. Casos de Uso - Tutor

O cadastro de curso é disponível para todos os usuários tutores, ele é realizado em

três etapas. Na primeira é informado os dados do curso como nome, carga horária total, categoria, uma breve descrição e também os módulos do curso.

The screenshot shows the 'Cadastrar curso' (Register course) page in the TutorEAD system. The page is divided into three tabs: 'Dados', 'Aulas', and 'Tarefas', with 'Dados' currently selected. The form includes several input fields: 'Nome do Curso' (Course Name), 'Descrição do curso' (Course Description), 'Carga Horária' (Hours), and 'Categoria' (Category). Below these fields is a section for adding modules, with 'Adicionar módulos' on the left and 'Módulos adicionados' on the right. The 'Adicionar' button is visible at the bottom left of the module section. A 'Próximo' (Next) button is located at the bottom right of the form. The page header includes the TutorEAD logo and the user's name 'Jose'.

Figura 9. Tela de cadastro dos cursos - Etapa 1

Na Figura 9 temos um exemplo de como é a tela para realizar o cadastro do curso, onde é possível observar todos dados solicitados na primeira etapa.

Na segunda etapa é efetuado o cadastro das aulas que irá contemplar cada módulo, sendo necessário informar o nome ou descrição da aula e também a carga horária. Nota-se que na etapa dois, os módulos disponíveis para o cadastro das aulas são os mesmo já cadastrados na etapa 1, então caso seja necessário realizar alguma alteração na criação do módulo é preciso voltar a etapa anterior. A cada aula adicionada o sistema mostrará a carga horária restante do curso, considerando a carga horária total informada na etapa anterior e a carga horária da própria aula adicionada, como evidenciado da Figura 12.

The screenshot shows the 'Cadastrar curso' (Register course) page in the TutorEAD system, now on the 'Aulas' (Classes) tab. The form is titled 'Adicionar Aulas (Lets Programing)'. It includes input fields for 'Digite o nome ou descrição da aula' (Enter the name or description of the class) and 'Digite a carga horária da aula' (Enter the class hours). An 'Adicionar' button is located below these fields. On the right side, the 'Aulas adicionadas' (Added classes) section shows a list of classes, with the first entry being 'Aula 1 (1 Hrs)'. A 'Próximo' (Next) button is located at the bottom right of the form. The page header includes the TutorEAD logo and the user's name 'Jose'.

Figura 10. Tela de cadastro dos cursos - Etapa 2

Na última etapa, mostrada na Figura 11, o tutor consegue adicionar conteúdos a

cada aula, podendo enviar tarefas tais como apresentações, vídeos ou somente textos. Da mesma forma que nas outras etapas, as aulas disponíveis para envio de atividade são as cadastradas na etapa anterior. As atividades cadastradas ficarão disponíveis na tela do curso acessada pelos Alunos, referenciando o módulo e aula respectiva a cada uma.

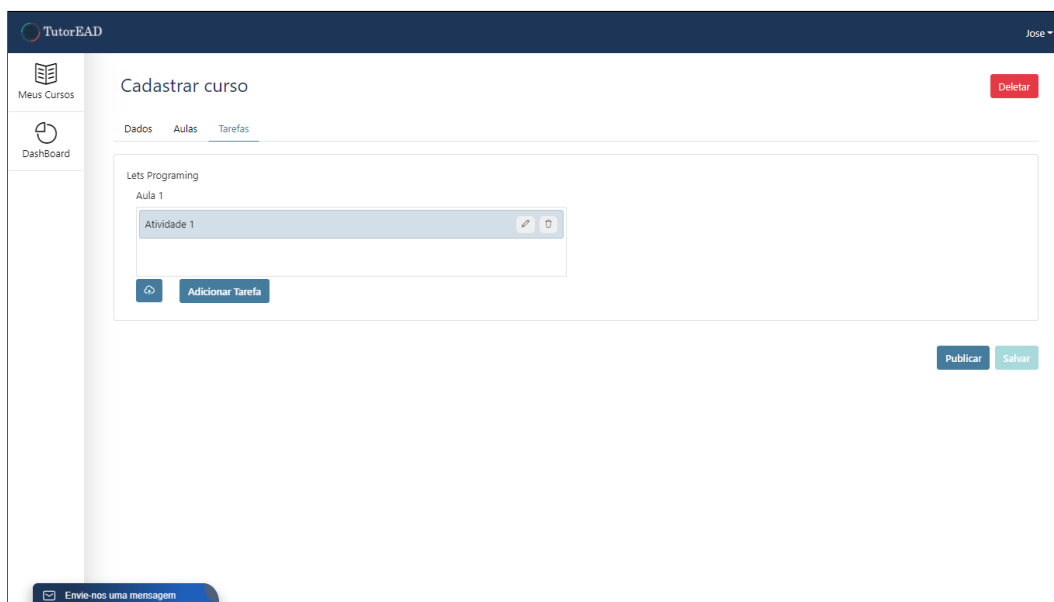


Figura 11. Tela de cadastro dos cursos - Etapa 3

Após o preenchimento de todos os dados, módulos e aulas o tutor poderá salvar ou publicar o curso para que o curso fique disponível na lista de cursos disponíveis, podendo ser visualizado pelos usuários alunos.

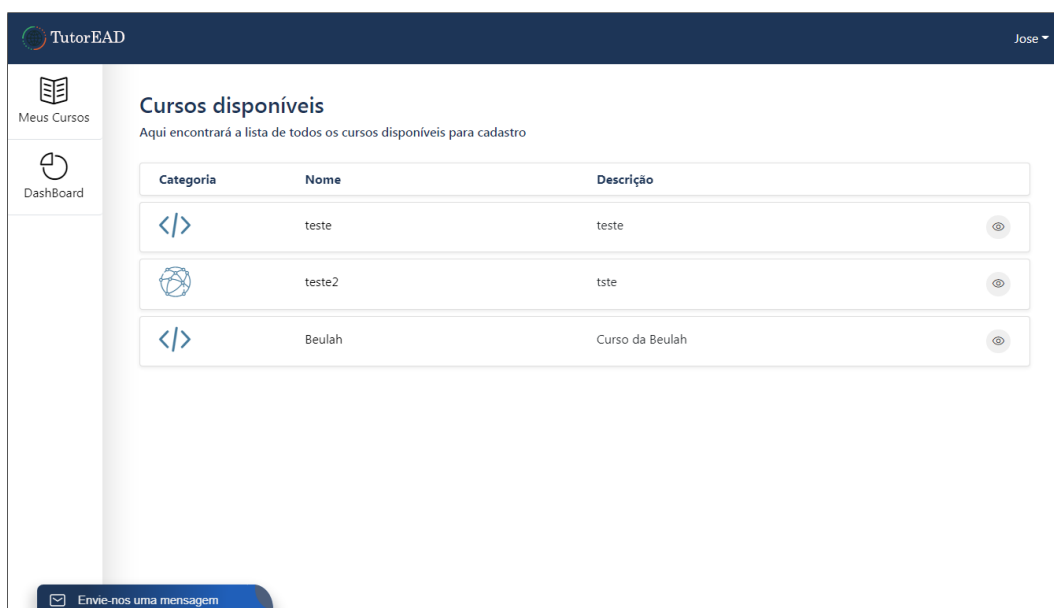


Figura 12. Tela de listagem dos cursos

Os tutores podem realizar mudanças em um curso, na parte que desejarem, po-

endo ser um módulo todo, ou apenas uma atividade. Para realizar a edição o tutor terá que acessar o curso através da lista de cursos disponíveis e ao invés de ser levado até a tela do curso, ele é redirecionado para a página de edição, tela similar a de cadastro, porém com todas as informações já preenchidas anteriormente.

No desenvolvimento desta tela observou-se um problema que não havia sido premeditado. Como as telas foram feitas em cima de uma mesma *view*, o modelo com as informações do curso seria enviado para o servidor somente quando o usuário clicasse no botão salvar, sendo assim ao carregar os conteúdos das aulas o *upload* seria feito com o todo o resto do modelo, o que acabava por consumir a memória do navegador do cliente, e dependendo do tamanho do arquivo, até mesmo parando completamente a navegação do usuário.

Para solucionar este problema, foi criada uma verificação entre a passagem da tela de módulos para a tela de aulas, a verificação consiste em conferir se o curso está salvo na base de dados e a seção atual não passa de uma edição e se é de fato uma edição, conferi-se também se foram criadas mais aulas ou módulos para o curso. Se as verificações passarem, o curso é salvo antes de passar para a tela de aulas, o modelo do curso é atualizado no lado do cliente e a tela é aberta.

Assim quando um *upload* for feito para uma aula específica, já é obtido o identificador daquela aula na base e é possível fazer o *upload* antes de clicar no botão de salvar, liberando assim a possibilidade de limpar o conteúdo do *upload* no lado do cliente, liberando a memória.

3.2.7. Módulo Aluno

Este módulo consiste em gerenciar as permissões do ator aluno. Com o nível de usuário aluno é possível visualizar a lista dos cursos disponíveis e realizar a matrícula para participar do curso, portanto o aluno terá uma interface para ter acesso a todo o conteúdo do curso disponibilizado e também para acesso as notas e correções efetuadas pelo tutor do curso. O aluno não tem acesso ao cadastro e edição do curso.

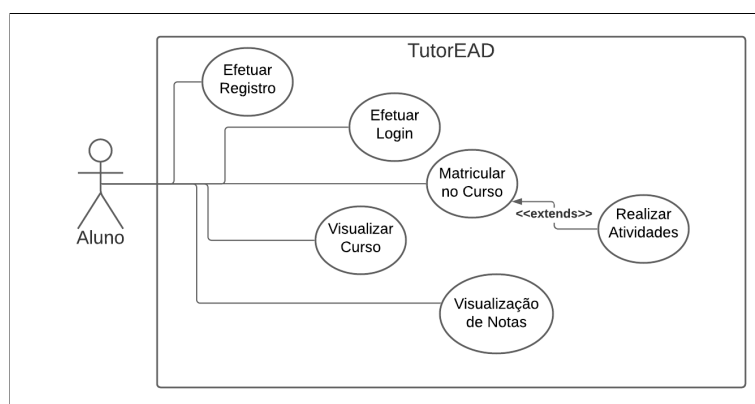


Figura 13. Casos de Uso - Aluno

Para a visualização dos cursos disponíveis, basta o aluno acessar a página através do ícone presente na *sidebar* do *layout* da ferramenta, na listagem é apresentado os cursos,

mostrando algumas informações, sendo a categoria, o nome e a descrição cadastrada na primeira etapa do cadastro efetuado pelo ator tutor. Ao selecionar um curso é possível fazer a matrícula no mesmo, e então ele é listado na página Meus Cursos, que pode ser acessada através do ícone na *sidebar*. Quando se é selecionado um curso listado em “Meus Cursos”, o aluno é redirecionado diretamente a tela do curso, tendo acesso a todo o material cadastrado.

Ao selecionar um módulo, ele é estendido mostrando todas as aulas e suas atividades relacionadas, as atividades aparecerão no quadro de conteúdo. Conforme o aluno for concluindo as atividades e avançando as aulas a barra de percentual irá se ajustando, para se ter o controle do progresso do curso.

3.2.8. Módulo Administrador

Os atores administradores terão todas as funções que estarão disponíveis tanto para os tutores quanto para os alunos, e possuem a autonomia para transformar um usuário aluno para o nível tutor para ser possível fazer o cadastro de cursos, portanto sempre que um usuário cadastrado desejar ser um tutor é necessário entrar em contato com os administradores para realizar esta solicitação.

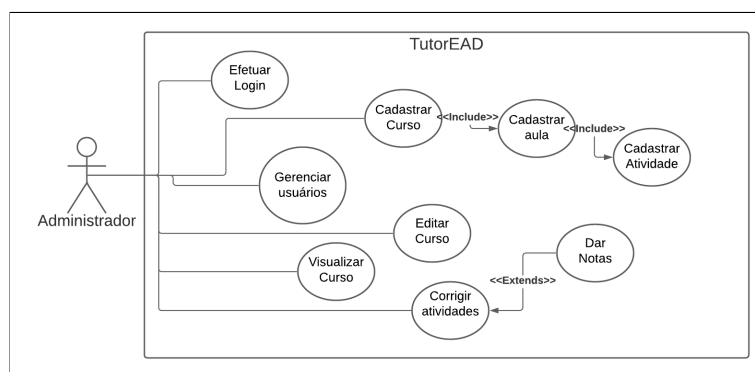


Figura 14. Casos de Uso - Administrador

3.3. Resultados

Após a conclusão do desenvolvimento da ferramenta foi aplicada a avaliação da mesma, com docentes dos cursos de tecnologia do campus Lages do Instituto Federal de Santa Catarina de forma voluntária, a pesquisa contou com 7 participantes que realizaram testes nas funcionalidades desenvolvidas. Os testes aconteceram de forma prática, onde o usuário fez a utilização da plataforma de forma geral, para poder verificar a sua experiência e os resultados apresentados pelo sistema.

Posteriormente aos testes foi disponibilizado um formulário *online*, onde foi possível coletar as opiniões dos voluntários. A avaliação foi realizada dividida entre os módulos presentes na plataforma, onde cada um foi avaliado de forma individual. Para a confecção do formulário foram considerados pontos importantes relacionados a experiência dos usuários como a interface/*layout*, a usabilidade, e a apresentação dos dados.

A seguir são apresentados os resultados da avaliação referente a tela de *login* e registro da plataforma. É possível observar um resultado satisfatório em relação às respostas obtidas. Em relação a interface e o *layout* ser amigável e atrativo, 71,4% das respostas classificaram como bom e 28,6% como excelente, como é possível observar no gráfico da Figura 15.

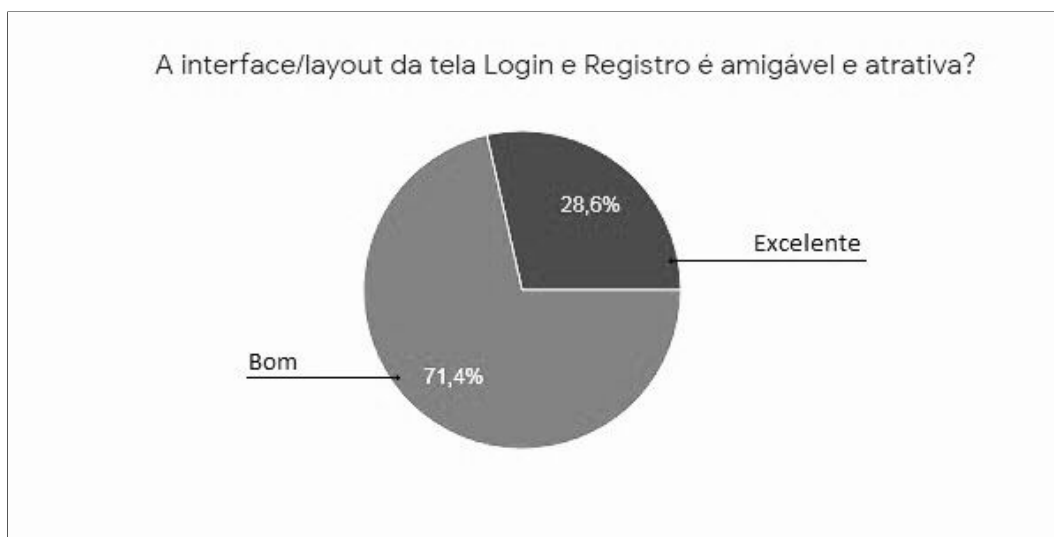


Figura 15.

Referente a tela ser intuitiva, 57,1% classificaram como bom, 28,6% como excelente, e 14,3% como regular, demonstrando também um resultado satisfatório.

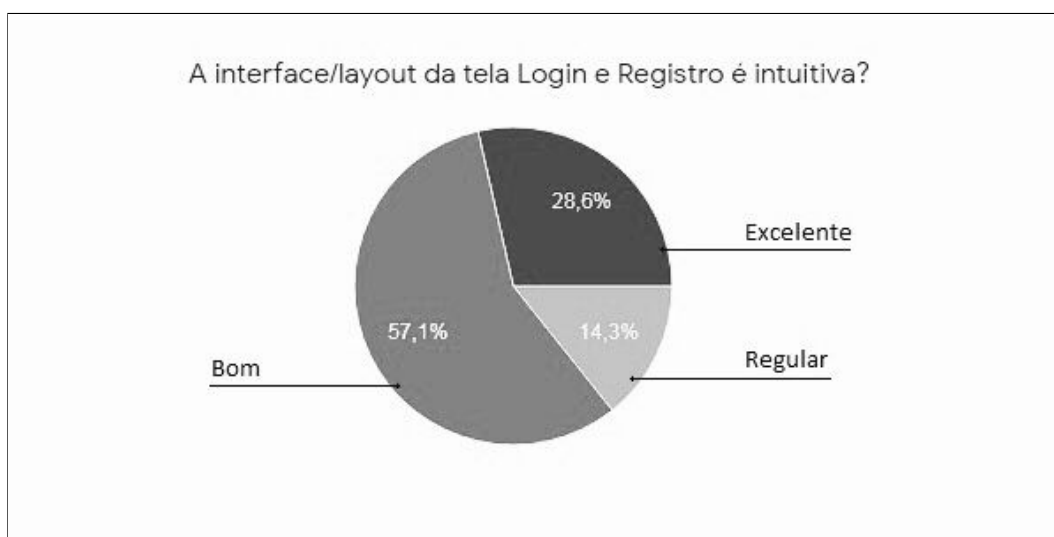


Figura 16.

Tratando-se da apresentação dos dados, cerca de 71,4% classificaram como bom e 28,6% como excelente. É possível observar nos gráficos que esta funcionalidade, foi bem avaliada com a maioria de suas classificações como bom ou excelente.



Figura 17.

Os gráficos representados nas próximas Figuras mostram os resultados da avaliação referente a funcionalidade de cadastro e edição de cursos, onde também foi possível observar uma avaliação positiva. Em relação a interface e *layout* ser agradável obtiveram-se 85,7% das repostas como bom e os outros 14,3% como excelente.

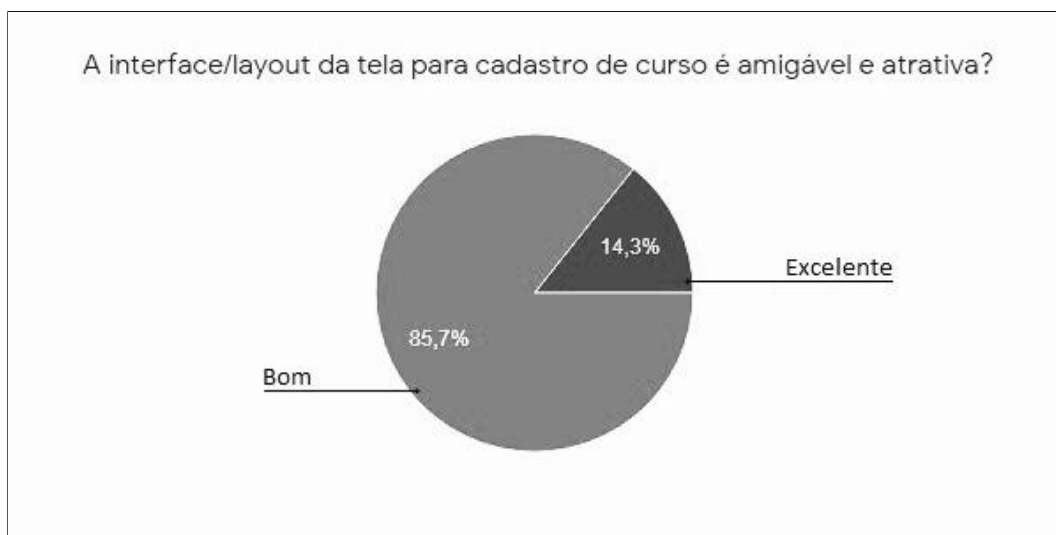


Figura 18.

Na Figura 19 é mostrado referente a intuitividade da página de cadastro e edição de cursos 71,5% classificaram como bom ou excelente e 28,6% como regular.

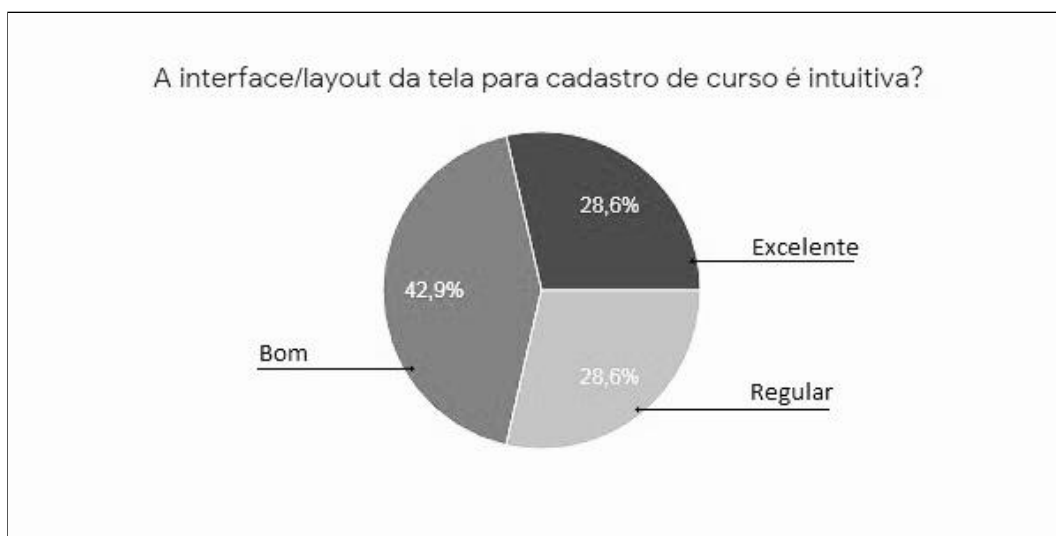


Figura 19.

Associado a apresentação e solicitação de dados, 57,2% dos avaliadores concordaram ser bom ou excelente e 42,9% ser regular.

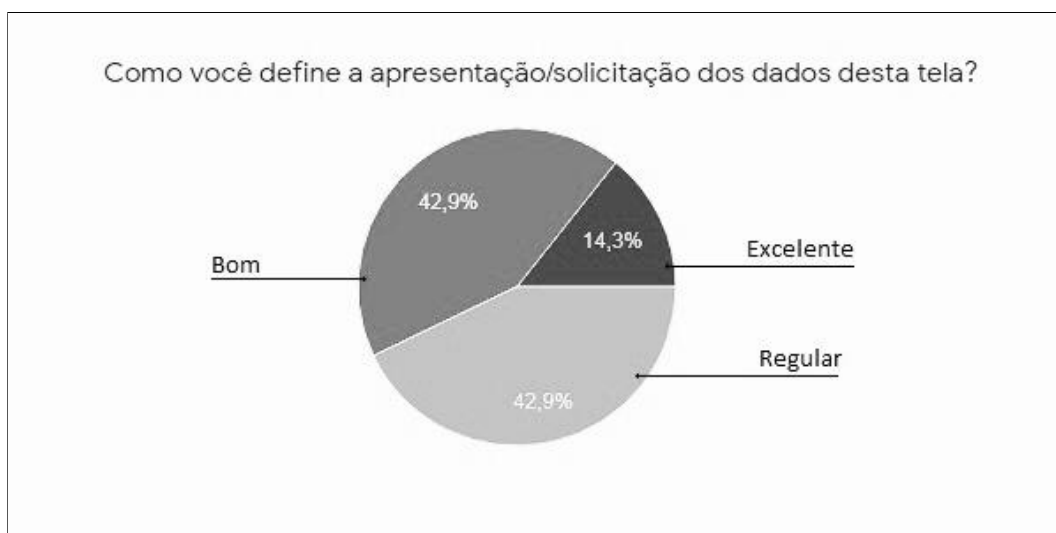


Figura 20.

Para esta funcionalidade foram acrescentadas duas perguntas com o intuito de avaliar se foi possível executar as funcionalidades apresentadas de forma prática, os gráficos destas perguntas são apresentados na Figura 21 e na Figura 22. Para o cadastro de curso, 42,9% classificaram como excelente a praticidade, 28,6% classificaram como bom e regular.

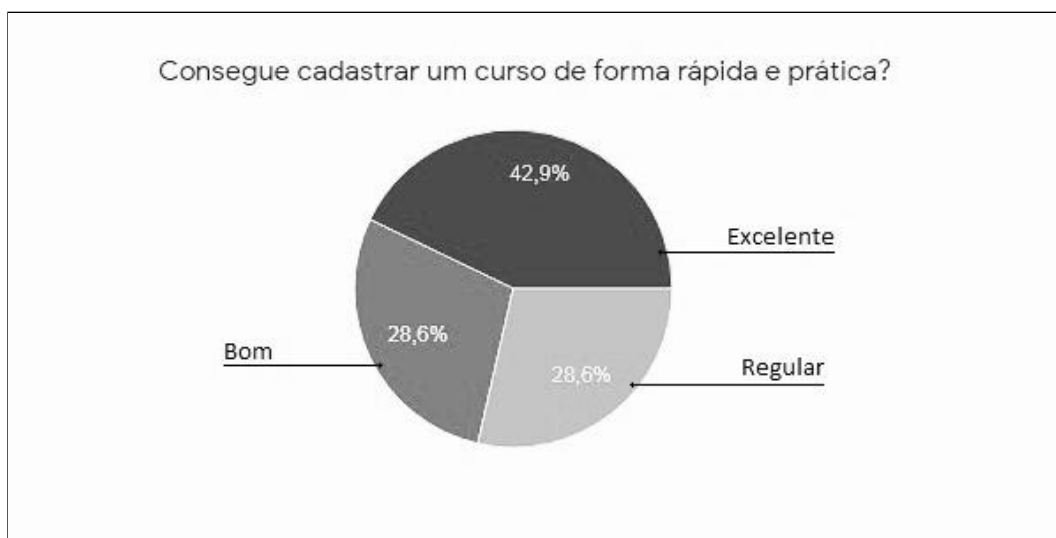


Figura 21.

Já para edição de cursos, 42,9% marcaram como excelente e bom e os outros 14,3% como regular.



Figura 22.

Por último avaliou-se a página Meus Cursos disponível para os tutores da plataforma. A interface deste módulo foi classificada 57,1% como bom, 28,6% como excelente e 14,3% como regular.

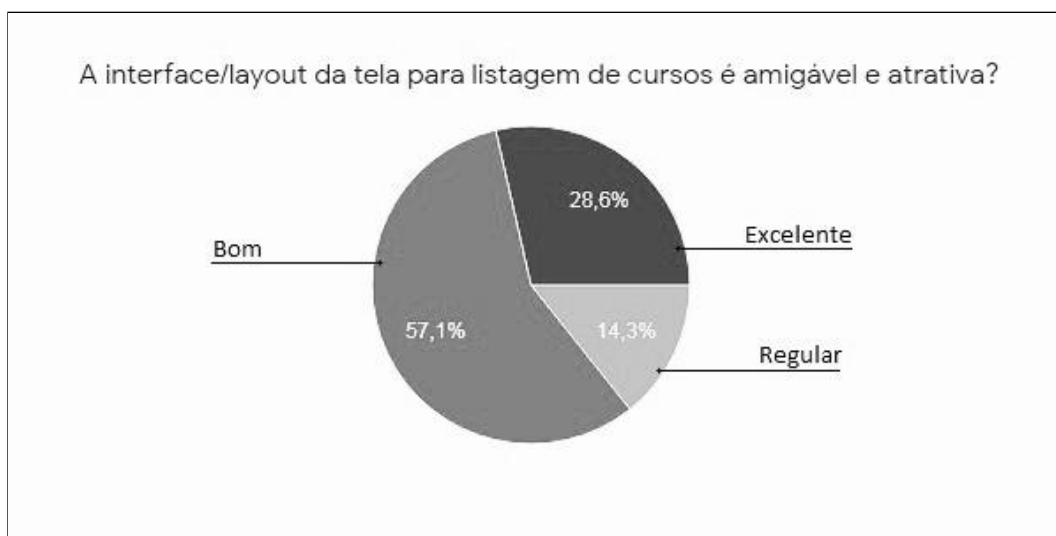


Figura 23.

Em relação à tela ser intuitiva 71,4% assinalaram como bom e 28,6% como excelente.

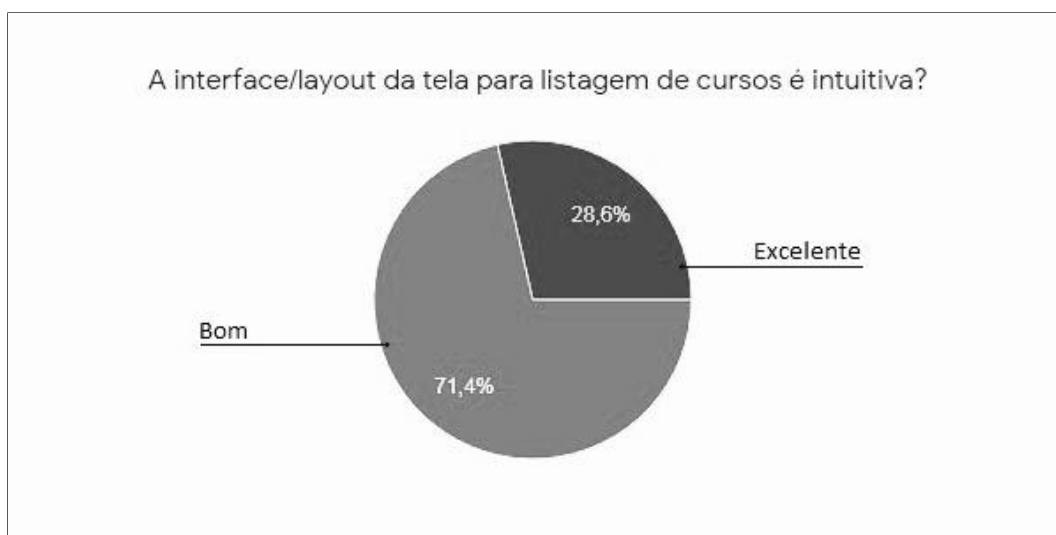


Figura 24.

A apresentação de dados ficou classificada com 42,9% como bom, 28,6% como excelente e os outros 28,6% como regular.

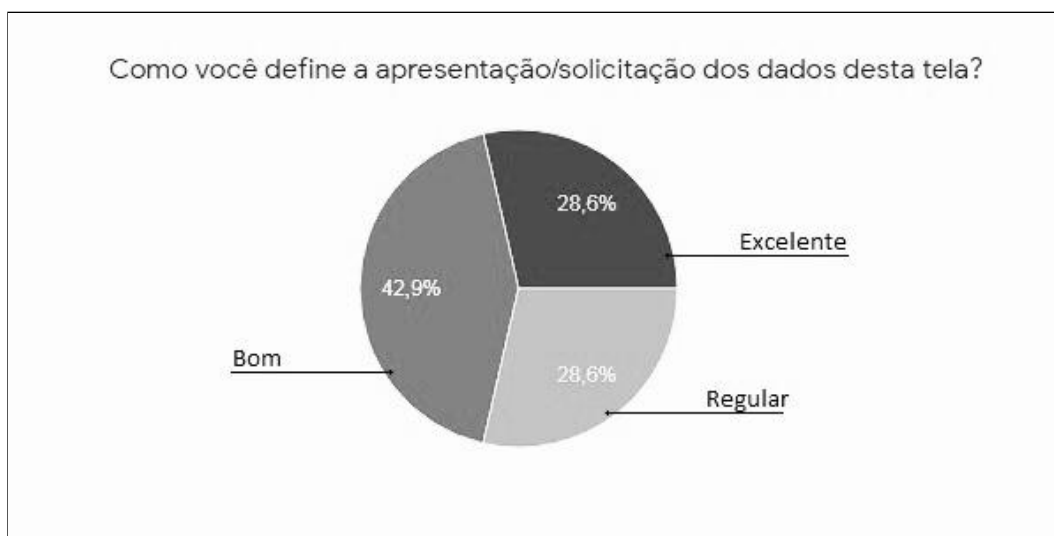


Figura 25.

Nesta funcionalidade também houve a adição de uma questão referente ao acesso de forma rápida e prática a funcionalidade, onde 71,4% concordaram em ser excelente e 28,6% como bom.

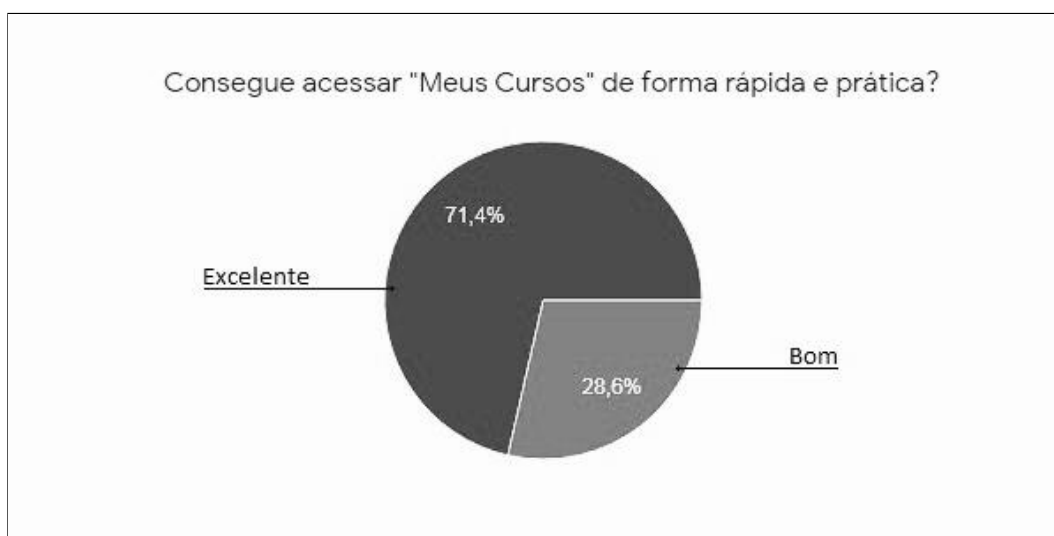


Figura 26.

Ao fim do formulário foi disponibilizado um campo para ser realizadas críticas ou sugestões referentes a plataforma. As respostas coletadas através desta questão foram verificadas com o intuito de serem implementadas na ferramenta, para uma melhor experiência de usuário, atendendo a opinião da comunidade local.

4. Conclusões

O trabalho apresentou o desenvolvimento de uma plataforma para organização sistemática de material didático, para auxiliar no processo de ensino-aprendizagem, onde professores

podem oferecer seus cursos de forma gratuita e *online* para alunos de uma comunidade. Como o ensino a distância vem se tornando cada vez mais presente, a plataforma pode auxiliar na amplificação do acesso a esse método de ensino.

Para desenvolver uma plataforma menos genérica realizaram-se estudos referente a experiência de usuário para se atribuir ao sistema, onde foi possível se obter uma experiência satisfatória no uso do sistema. Foi utilizado de técnicas e regras de *design* para realizar a construção das telas e *layouts*. O uso dessas técnicas tornou o sistema menos genérico, proporcionando uma melhor usabilidade para os usuários.

A plataforma foi avaliada através de testes práticos realizados pelos docentes do campus do IFSC Lages. Os resultados da avaliação obtidos por meio do formulário disponibilizado para os avaliadores, mostram que a plataforma está com conceito de regular a excelente.

Em relação aos trabalhos futuros, será verificado as sugestões citadas pelos avaliadores durante o questionário de avaliação da plataforma, implementando as correções necessárias para a melhoria da experiência de usuário da plataforma. Além disso, será realizado o desenvolvimento de novas funcionalidades dentro da plataforma, como a apresentação do *dashboard* para acompanhamento dos alunos.

Referências

- Alves, L. (2001). Educação a distância: conceitos e história no Brasil e no mundo. *Revista Brasileira de Aprendizagem Aberta e a Distância (RBAAD)*, 10.
- Araújo, M. (2013). Ead em tela: docência, ensino e ferramentas digitais. *Revista Brasileira de Linguística Aplicada*, 14:735–742.
- Baranauskas, M. e Rocha, H. (2000). *Design e avaliação de interfaces humano-computador*. UNICAMP.
- Ferris, K. e Zhang, S. (2016). A framework for selecting and optimizing color scheme in web design. *49th Hawaii International Conference on System Sciences*.
- Gamma, E., Helm, R., Johnson, R., e Vlissides, J. (2000). *Padrões de Projeto: Soluções Reutilizáveis de Software Orientado a Objetos*. Bookman.
- Garrido, F. A., do Rêgo, B. B., e de Souza Matos, E. (2018a). Design instrucional orientado a artefatos: uma abordagem participativa e distribuída.
- Garrido, F. A., do Rêgo, B. B., e de Souza Matos, E. (2018b). Moodle como ambiente mooc: orientações para o redesign de interação. *CINTED-UFRGS*.
- Garrido, F. A., do Rêgo, B. B., Maciel, R. S. P., e de Souza Matos, E. (2019). Uma abordagem de design para mooc: um mapeamento sistemático da articulação entre design instrucional e de interação.
- Gomes, F., Varanis, H., Giusepin, L., e Oliveira, E. (2015). Elementos que influenciam a experiência do usuário na utilização de web sites. *Intercom – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação*.
- Grilo, A. (2019). *Experiência do Usuário em Interfaces Digitais*. SEDIS-UFRN, Rio Grande do Norte.
- Hack, J. R. (2011). *Introdução à educação a distância*. LLV/CCE/UFSC, Florianópolis.
- Krasner, G. e Pope, S. (1988). A cookbook for using model-view-controller user interface paradigm in smaltalk-80.

- Morais, A., Oliveira, K., Santos, J., e Oliveira, R. (2020). Proposta de uma ferramenta tecnológica focada no auxílio na educação de jovens e adultos. *IX Congresso Brasileiro de Informática na Educação*.
- Moran, J. M. (2000). O que é educação a distância. *ECA-USP*.
- Renz, C., Euzebio, L., e Bohn, J. (2019). Fazendo e aprendendo: a criação de um software didático como ferramenta complementar no processo de ensino e aprendizado em contabilidade. *CBIE 2019*.
- Rhis, D. H. (2018). Desenvolvimento de uma plataforma educacionao para disseminação do conhecimento. *Universidade Federal dos Vales do Jequitinhonha e Mucuri*.
- Rogers, Y., Sharp, H., e Preece, J. (2013). *Design de Interação: Além da Interação Humano-Computador*. Bookman.
- Schlemmer, E. e Fagundes, L. (2000). Uma proposta para avaliação de ambientes virtuais de aprendizagem na sociedade em rede. *Revista Brasileira de Aprendizagem Aberta e a Distância (RBAAD)*, 2.
- Sommerville, I. (2011). *Engenharia de Software*, volume 9. Pearson Prentice Hall.
- Sutherland, J. V. (2019). *Scrum: a arte de fazer o dobro do trabalho na metade do tempo*. Sextante.
- Wazlawick, R. S. (2009). *Metodologia de Pesquisa para Ciência da Computação*. Elsevier.
- Zanella, L. C. H. (2013). *Metodologia de pesquisa*. UFSC, Florianópolis: Departamento de Ciências da Administração/UFSC, 2ª edition.