

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS FRANCISCO BELTRÃO
CURSO SUPERIOR DE LICENCIATURA EM INFORMÁTICA**

MARCOS MARCOLIN

**PROTÓTIPO DE SOFTWARE WEB DE APOIO AO ENSINO
PRESENCIAL COM FOCO NA INTERFACE DO USUÁRIO**

TRABALHO DE CONCLUSÃO DE CURSO

**FRANCISCO BELTRÃO
2017**

MARCOS MARCOLIN

**PROTÓTIPO DE SOFTWARE WEB DE APOIO AO ENSINO
PRESENCIAL COM FOCO NA INTERFACE DO USUÁRIO**

Trabalho de Conclusão de Curso de graduação, apresentado à disciplina de Trabalho de Conclusão de Curso, do Curso Superior de Licenciatura em Informática, da Universidade Tecnológica Federal do Paraná, Câmpus Francisco Beltrão, como requisito parcial para obtenção do título de Licenciado.

Orientadora: Profa. Dra. Maici Duarte Leite

**FRANCISCO BELTRÃO
2017**

MARCOS MARCOLIN

**PROTÓTIPO DE SOFTWARE WEB DE APOIO AO ENSINO
PRESENCIAL COM FOCO NA INTERFACE DO USUÁRIO**

Trabalho de Conclusão de Curso,
apresentado a Universidade Tecnológica
Federal – Campus Francisco Beltrão,
como parte das exigências para a
obtenção do título de Licenciado em
Informática.

UTFPR-FB, 31 de Outubro de 2017.

BANCA EXAMINADORA

Profa. Maici Duarte Leite (Orientadora) - UTFPR
Doutora em Ciência da Computação

Prof. Ademir Roberto Freddo (Convidado) - UFFS
Doutor em Engenharia Elétrica e Informática Industrial

Prof. Francisco A. F. Reinaldo (Presidente da Banca) - UTFPR
Doutor em Engenharia Electrotécnica e de Computadores

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso (ou Programa)”.

*Dedico este trabalho a minha família,
especialmente a minha Mãe Marilene
Aparecida Ubert Marcolin, ao meu irmão
Marcelo Marcolin e a minha namorada
Ágatha Mendes Castiglioni.*

AGRADECIMENTOS

Primeiramente, agradeço a instituição UTFPR Câmpus Francisco Beltrão por proporcionar todo o apoio necessário para a realização deste trabalho.

Agradeço imensamente a minha família, Mãe Marilene e meu Irmão Marcelo, que com toda humildade, simplicidade e honestidade ensinaram-me a ser uma pessoa de caráter e sempre incentivando-me a ir em busca de meus sonhos sem nunca passar por cima de ninguém, também por sempre estarem do meu lado em todos os momentos, apoiando-me a continuar com os estudos, vocês me ajudaram a realizar um sonho.

Agradeço a minha namorada Ágatha por ter entrado em minha vida e por acompanhar-me nesta jornada, pelo seu apoio, compreensão, paciência, dedicação e companheirismo em momentos que tive que abrir mão do lazer para estudar.

Agradeço ao meu Orientador do TCC I Prof. MSc. Marcos Mincov Tenório e a minha Orientadora do TCC II Profa. Dra. Maici Duarte Leite, por todos os ensinamentos repassados a mim de forma clara e objetiva, por me guiarem nesta etapa, além das experiências profissionais e pessoais, as quais levarei para a vida toda.

Agradeço aos Professores que compuseram a banca deste TCC, Prof. Dr. Ademir Roberto Freddo e Prof. Dr. Francisco Antônio Reinaldo Fernandes por suas presenças e avaliações. Também agradeço a todos os Docentes da COLIN aos quais tive o prazer de ser aluno durante esta graduação.

Agradeço a todos os meus colegas que convivi durante este tempo, que sempre apoiaram-me e alegraram-me todas as noites de aula, espero levar essa amizade para toda a vida.

Enfim, o meu mais sincero “muito obrigado” a todos as pessoas citadas acima, abraços a todos!

*Você não pode mudar o vento, mas pode
ajustar as velas do barco para chegar
onde quer.*

Confúcio

RESUMO

MARCOLIN, Marcos. **PROTÓTIPO DE SOFTWARE WEB DE APOIO AO ENSINO PRESENCIAL COM FOCO NA INTERFACE DO USUÁRIO**. 2017. 58 f. Monografia (Trabalho de Conclusão de Curso) - Curso Superior de Licenciatura em Informática, Universidade Tecnológica Federal do Paraná, Câmpus Francisco Beltrão. Francisco Beltrão, 2017.

Este trabalho consiste no desenvolvimento de um protótipo funcional do tipo Ambiente Virtual de Aprendizagem (AVA) para o apoio ao ensino presencial. O protótipo é focado na interface do usuário, com o objetivo de facilitar o seu uso por parte do aluno, visando o fácil acesso ao conteúdo disponibilizado pelo professor responsável do curso. Para o entendimento do desenvolvimento do protótipo, a pesquisa concentrou-se em *design* de interface e usabilidade do AVA. O Método utilizado escolhido foi a prototipação, por facilitar a alteração de elementos visuais e correção de erros antes de lançar o produto final. Fatores como educação, ensino-aprendizagem, interação humano-computador e *design* foram essenciais neste projeto. Através do desenvolvimento, o resultado foi a validação e apresentação do protótipo funcional de apoio ao ensino presencial.

Palavras-chave: *Design* de Interface, Usabilidade, Ambiente Virtual de Aprendizagem.

ABSTRACT

MARCOLIN, Marcos. **WEB SOFTWARE PROTOTYPE OF SUPPORT FOR PRESENCIAL TEACHING WITH FOCUS ON THE USER INTERFACE.** 2017. 58 f. Monograph (Work of Conclusion Course) - Graduation in Licenciatura em Informática, Federal University of Technology - Paraná, Campus Francisco Beltrão. Francisco Beltrão, 2017.

This project aims to develop a functional prototype of a Virtual Learning Environment (VLE) to support presential. The prototype were developed to be an user interface centered software aiming at student's ease of use, in order to facilitate their access to content provided by teachers or class tutors. To achieve the prototype's objective this project were based on VLE's interface design and usability. The engineering life cycle model used were Prototyping, mainly because it was originally conceived to be a faster method for creating prototypes and support the visual elements development by the ease of error correction. Education, teaching-learning process, human-computer interaction and design factors build the theoretical foundation of this project and, through this development, the result achieved consists in user's validation of the VLE functional prototype. We hope that this VLE, when completely deployed, support the presential.

Keywords: Interface Design, Usability, Virtual Learning Environment.

LISTA DE FIGURAS

Figura 1: Tela inicial do EasyPHP.....	31
Figura 2: Prototipação de Pressman.....	32
Figura 3: Diversos componentes do Bootstrap.....	35
Figura 4: Sistema de Grids.....	36
Figura 5: Carregamento do Bootstrap.....	36
Figura 6: Chamada do arquivo “meta.php”.....	37
Figura 7: Código HTML e CSS do Formulário para criar um Tópico.....	37
Figura 8: Código jQuery para mostrar e esconder elementos HTML.....	38
Figura 9: Ambiente do Fórum antes da interação do botão “CRIAR TÓPICO”.....	38
Figura 10: Ambiente do Fórum após a interação do botão “CRIAR TÓPICO”.....	39
Figura 11: Criação de um Novo Tópico no Fórum.....	40
Figura 12: Correspondência dos Ícones do Menu do Aluno.....	41
Figura 13: Ações para o autor de um Tópico.....	41
Figura 14: Consistência de Padrões.....	42
Figura 15: Prevenção de Erros.....	43
Figura 16: Breadcrumbs: Fluxo de Ações do Usuário.....	43
Figura 17: Máscaras de Campos em um Formulário.....	44
Figura 18: Design minimalista.....	45
Figura 19: Tela de Login.....	46
Figura 20: Perguntas mais Frequentes(FAQ).....	47
Figura 21: Interface Meu Painel.....	48
Figura 22: Interface de Cursos.....	48
Figura 23: Interface de Atividades.....	49
Figura 24: Interface de Materiais Oficiais da Disciplina.....	49
Figura 25: Interface de Alunos Matriculados.....	50
Figura 26: Interface do Fórum.....	50
Figura 27: Interface de um Tópico no Fórum.....	51
Figura 28: Interface Perfil do Aluno.....	51
Figura 29: Interface de Ambientes Responsivos.....	52
Figura 30: Casos de Uso do Aluno.....	58

LISTA DE QUADROS

Quadro 1: Tecnologias e Ferramentas Utilizadas.....	26
---	----

LISTA DE SIGLAS

AVA	Ambiente Virtual de Aprendizagem
CSS	Cascading Style Sheets
FAQ	Frequently Asked Questions
HTML	HyperText Markup Language
IHC	Interação Humano-Computador
JS	Javascript
PHP	Hypertext Preprocessor
TICs	Tecnologias De Informação e Comunicação
UTFPR	Universidade Tecnológica Federal do Paraná

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS.....	14
1.1.1 Objetivo Geral.....	14
1.1.2 Objetivos Específicos.....	14
1.2 JUSTIFICATIVA.....	15
1.3 ESTRUTURA DO TRABALHO.....	15
2 REFERENCIAL TEÓRICO.....	17
2.1 DESIGN DE INTERFACE E USABILIDADE.....	17
2.2 AMBIENTE VIRTUAL DE APRENDIZAGEM.....	21
2.3 TRABALHOS RELACIONADOS.....	24
3 MATERIAIS E MÉTODO.....	25
3.1 TECNOLOGIAS E FERRAMENTAS APLICADAS.....	25
3.1.1 Hypertext Preprocessor (PHP).....	26
3.1.2 Hypertext Markup Language (HTML).....	28
3.1.3 Cascading Style Sheets (CSS).....	28
3.1.4 Javascript (JS).....	29
3.1.5 Ambiente de desenvolvimento com EasyPHP Server.....	30
3.2 MÉTODO.....	31
4 RESULTADOS.....	34
4.1 ESCOPO DO SOFTWARE.....	34
4.2 IMPLEMENTAÇÃO DO PROTÓTIPO.....	35
4.3 VALIDAÇÃO DO PROTÓTIPO ATRAVÉS DAS 10 HEURÍSTICAS DE NIELSEN	39
4.3.1 Visibilidade do Estado do Sistema.....	39
4.3.2 Correspondência entre Sistema e Mundo Real.....	40
4.3.3 Controle e liberdade para o usuário.....	41
4.3.4 Consistência e Padronização.....	42
4.3.5 Prevenção de Erros.....	43
4.3.6 Reconhecimento em vez de memorização.....	43
4.3.7 Flexibilidade e eficiência de utilização.....	44
4.3.8 Design estético e minimalista.....	44

4.3.9 Ajude os usuários a reconhecer, diagnosticar e recuperar erros.....	45
4.3.10 Ajuda e Documentação.....	46
4.4 APRESENTAÇÃO DO PROTÓTIPO.....	47
5 CONSIDERAÇÕES FINAIS.....	53
5.1 TRABALHOS FUTUROS.....	54
6 REFERÊNCIAS.....	55
7 APÊNDICES.....	58

1 INTRODUÇÃO

A importância das Tecnologias de Informação e Comunicação (TICs) para a prática educativa estão em foco em instituições de ensino e pesquisas atuais, tendo como objetivo fornecer suporte ao ensino presencial. Em paralelo a isto foi observado que os modelos tradicionais, onde o aprendizado é centrado no professor, parecem não atrair nem motivar os estudantes. Assim, a utilização de TICs torna-se essencial para mudar tais cenários e favorecer a inserção do estudante no centro do seu processo de aprendizagem (NORMAN; SPOHRER, 1996).

Neste novo modelo, os Ambientes Virtuais de Aprendizagem (AVAs) apresentam-se como uma possibilidade de permitir espaços virtuais em que o aluno tenha a disposição materiais didáticos, atividades, fóruns, interações com professores, interações com colegas, materiais colaborativos, e inúmeras outras funcionalidades (ALVES *et al.*, 2009).

Porém, não é o suficiente apenas inserir um AVA a espera de resultados promissores. Krahe *et al.* (2006) argumentam que existe uma grande expectativa que as novas tecnologias sejam a solução, se bem empregadas podem mudar e qualificar as práticas pedagógicas, mas somente tê-las na escola e utilizá-las de qualquer maneira não é garantia de melhoria do processo de ensino aprendizagem.

A forma de utilização das TICs variam para cada usuário, alguns com facilidades, outros com dificuldades. Por isso torna-se necessário o desenvolvimento de *softwares* planejados através da Interação Humano-Computador (IHC). Preece (1994) afirma que IHC está relacionada com o *design* de sistemas computacionais que apoiem as pessoas de forma que possam conduzir suas atividades de forma produtiva e com segurança.

A IHC então apresenta-se como um fator determinante na construção de ambientes educacionais, e por consequência fator determinante na construção de

Ambientes Virtuais de Aprendizagem (Mozzaquatro e Medina 2008). Os autores supracitados, ainda afirmam que “a interface destes sistemas deve ser amigável e intuitiva, bem como, facilitar o seu uso e diminuir o processo exaustivo da busca de acesso à informação pelo usuário”.

Porém, ao observar os AVAs tradicionalmente usados nas instituições de ensino, verifica-se que nem sempre oferecem uma interface, usabilidade e *design* adequados para alunos e professores realizarem suas atividades com segurança e satisfação. Nestes contextos nota-se que a IHC é deixada de lado enquanto os esforços residem nas ações e funcionalidades do ambiente.

É nesta lacuna que este trabalho se desenvolveu, propondo o *design* de um protótipo funcional de um AVA destinado ao apoio do ensino presencial. O diferencial desta proposta residiu na construção do ambiente baseando-se inicialmente nos conceitos e métodos de IHC, utilizando como base práticas para construção de interfaces.

Com esta construção foi possível validar o protótipo através de métodos definidos na Literatura, especialmente atendendo as 10 Heurísticas de Nielsen para evitar erros de Usabilidade.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

- Construir uma interface para um Ambiente Virtual de Aprendizagem por meio de conceitos e técnicas de IHC, com validação na teoria de Nielsen (1994).

1.1.2 Objetivos Específicos

- Estudar a literatura especializada sobre AVA a fim de suporte para identificar seus requisitos e possíveis problemas de usabilidade;

- Relacionar conceitos de IHC, *design* de interface e usabilidade;
- Analisar as melhores práticas na construção de um AVA para aplicar na proposta;
- Construir um protótipo funcional do *software* AVA proposto para estruturar uma avaliação;
- Validar a interface e usabilidade atendendo as 10 Heurísticas de Nielsen (1994).

1.2 JUSTIFICATIVA

Para um AVA cumprir com seus objetivos, necessita-se uma interface para viabilizar os processos de comunicação entre o usuário e máquina. Souza e Costa (2006) afirmam que em um projeto de *software* que envolva IHC, a participação do usuário no projeto de interface é fundamental para o sucesso do sistema.

Desta forma, percebe-se que a interface é o primeiro contato do usuário com o *software*. SALLES et. al (2006) afirmam que a interface é o elo entre o usuário e computador, se esse contato não for amigável, o usuário poderá desistir de usar.

Entretanto, é desafiador construir interfaces que ajudem o usuário encontrar alguma informação de forma facilitada, e agravam-se tais desafios ao tratar de ambientes educacionais, de tal forma que seu mal uso pode acarretar em problemas no processo de ensino-aprendizagem.

Justifica-se, então, dentro de um AVA, a importância de uma interface interativa e amigável para a realização de tarefas, tanto para o usuário aluno, quanto ao usuário professor. Com isto, será possível obter contribuições para todo o processo de ensino-aprendizagem, favorecendo a utilização de um AVA através de um desenvolvimento da interface focada no usuário.

1.3 ESTRUTURA DO TRABALHO

O Capítulo 1 é composto pela parte introdutória do trabalho, objetivos e sua justificativa. O Capítulo 2 é destinado ao referencial teórico do trabalho, sendo *Design* de Interface e Usabilidade, e Ambiente Virtual de Aprendizagem e trabalhos relacionados. Já O Capítulo 3 detalha os materiais e métodos utilizados para o desenvolvimento do protótipo. Posteriormente, o Capítulo 4 mostra os resultados. E por fim, O Capítulo 5 conclui o trabalho.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os conteúdos e conceitos necessários para o desenvolvimento e entendimento deste trabalho. São detalhados os temas: *design* de interface e usabilidade e ambiente virtual de aprendizagem.

2.1 DESIGN DE INTERFACE E USABILIDADE

Primeiramente, é necessário entender ao que refere-se o termo *design*. Segundo BENYON (2011, p.5):

“O termo ‘*design*’ refere-se tanto ao processo criativo de especificar algo novo quanto às representações que se produzem durante este processo. Portanto, para projetar um site, por exemplo, o *design* produzirá e avaliará vários *designs*, como o *design* do *layout* da página, o do esquema de cores, o dos gráficos e o da estrutura como um todo.”

O Autor supracitado ressalta que um *design* tem que ser visualizado para ajudar um *designer* clarear as próprias ideias quanto para que as pessoas possam avaliá-lo.

A Interface é constituída pelas seguintes partes, as quais os usuários têm contato: física, perceptiva ou conceitualmente:

- **fisicamente:** podemos interagir com um dispositivo apertando botões ou movimentando alavancas e o dispositivo interativo pode responder fornecendo retorno através da pressão do botão ou alavanca;
- **perceptivamente:** o dispositivo exhibe coisas em uma tela que podemos ver, ou emite sons que podemos ouvir;
- **conceitualmente:** interagimos com um dispositivo tentando concluir o que ele faz e o que deveríamos estar fazendo. O dispositivo fornece mensagens e outros indicadores feitos para nos ajudar nesse sentido.

Portanto, a interface de um sistema é o meio pelo qual acontece o diálogo entre uma aplicação e o humano. Pressman (2011) define interface de forma parecida com especialistas em usabilidade:

“A interface do usuário é indiscutivelmente o elemento mais importante de um produto. Se a interface for mal projetada, a capacidade de o usuário aproveitar todo o poder computacional e conteúdo de informações de uma aplicação pode ser seriamente afetada. Na realidade, uma interface fraca pode fazer com que uma aplicação, em outros aspectos bem projetada e solidamente implantada, falhe.” (PRESSMAN, 2011)

Desta maneira, neste trabalho quando é citado *design* de interface, entende-se por um processo que constrói telas que podem ser utilizadas por usuários para navegar entre páginas de um *software*.

Um outro ponto crucial em uma interface, é a usabilidade, que assegura que o *software* seja fácil de usar, eficiente e agradável, de acordo com a perspectiva do usuário (PREECE; ROGER; SHARP, 2005). Estes autores definiram metas de usabilidade para um *software* atingir seu objetivo, sendo:

1. **eficácia no uso:** corresponde ao quanto o sistema é bom para fazer o que se espera dele;
2. **eficiência no uso:** corresponde à maneira como o sistema auxilia os usuários na realização das tarefas;
3. **fácil de lembrar como usar:** corresponde à facilidade de lembrar como utilizar o sistema;
4. **fácil de entender:** corresponde ao quão fácil é aprender a usar o sistema;
5. **de boa utilidade:** corresponde à medida na qual o sistema propicia o tipo certo de funcionalidade, de maneira que os usuários possam realizar aquilo de que precisam ou que desejam;
6. **seguro no uso:** corresponde a proteger os usuários de situações perigosas e indesejáveis.

A partir da última década, os usuários tornaram-se mais exigentes com a tecnologia utilizada, por isso as empresas estão investindo cada vez mais em produtos com um *design* sofisticado, fáceis de utilizar, rápidos e que tenham funcionalidade. Porém, nem todos sabem que estas características fazem parte da usabilidade de *software*. Segundo a NBR ISO/IEC 9126-1 (2003) usabilidade é a

capacidade do produto de *software* de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições especificadas.

SHACKEL (1991) considera a usabilidade como a capacidade de um sistema ser usado facilmente e com eficiência pelo usuário. Já NIELSEN (1990) e SANTOS (2000) definem de forma semelhante, dividindo o conceito de usabilidade em itens como facilidade e eficiência.

Em *softwares* educacionais não é diferente, o usuário quer utilizar um AVA e sentir-se entusiasmado a utilizar novamente quando possível. Porém, nos AVAs disponíveis e utilizados atualmente há uma dificuldade por parte de alunos e professores em relação a interface. Diversos autores já relataram seus problemas na literatura, incluindo o *Moodle*, um dos AVAs mais utilizados no mundo (Magalhães et. al 2010).

“Vamos pegar como exemplo o *Moodle*, por ser gratuito é largamente utilizado tanto no âmbito educacional como empresarial. É uma plataforma de aprendizagem com mais de 10 anos de mercado que inclui milhares de plugins e funcionalidades que podem ser exploradas pedagogicamente. Seus problemas de usabilidade concentram-se nas interfaces, ainda que as Instituições de Ensino se desdobram para oferecer cursos de qualidade, existe ainda a falta de profissionais capacitados e especializados na sua customização, resultando em layouts complexos.” (IVO, 2015).

Em outro estudo realizado para identificar problemas de uso do *Moodle* através de alunos, tutores e professores com foco em usabilidade, o resultado encontrado foi um problema ainda maior encontrado no ambiente:

“O maior problema de usabilidade encontrado no ambiente Moodle está relacionado ao uso das ferramentas que estão inseridas em locais de difícil acesso, dificultando e transformando o percurso cognitivo dos usuários, impossibilitando, muitas vezes, o acesso direto à ferramenta por alunos, tutores e professores.” (UFAL, 2011)(BITTENCOURT; BITTENCOURT; SANTOS, 2011).

É possível que, no desenvolvimento do *Moodle* ou de outras ferramentas educacionais que apresentam os mesmos problemas, não se tenha priorizado o *design* da interface e a usabilidade. Nota-se que os autores ressaltam a importância deste projeto bem realizado.

Nielsen (1994) aponta que um projeto do *software*, deve considerar a avaliação do *design* da interface. Sendo assim, um projeto deve ser construído para atender as 10 Heurísticas de Usabilidade de Nielsen, que são itens que ajudam a realizar um diagnóstico da usabilidade do *software*, facilitando ao *designer* projetar uma melhor experiência de uso e tornando as interações de fácil entendimento, sendo:

1. **Visibilidade do Estado do Sistema:** o sistema deve sempre manter os usuários informados sobre o que está acontecendo, através de *feedback* adequado dentro de um prazo razoável;
2. **Correspondência entre Sistema e Mundo Real:** o sistema deve falar o idioma dos usuários, com palavras, frases e conceitos familiares para o usuário;
3. **Controle e liberdade para o usuário:** os usuários muitas vezes escolhem uma função por engano e precisarão de uma “saída de emergência” claramente marcada para deixar o estado indesejado;
4. **Consistência e Padronização:** os usuários não devem ter que se perguntar se diferentes palavras, situações ou ações significam a mesma coisa;
5. **Prevenção de Erros:** melhor do que boas mensagens de erro é um *design* cuidadoso que impede que um problema ocorra;
6. **Reconhecimento em vez de memorização:** minimize a carga de memória do usuário, tornando visíveis objetos, ações e opções;
7. **Flexibilidade e eficiência de utilização:** muitas vezes pode acelerar a interação para o usuário especializados, de tal forma que o sistema pode servir tanto para usuários inexperientes e experientes;
8. **Design estético e minimalista:** os diálogos não devem conter informações irrelevantes ou raramente necessárias;

9. Ajude os usuários a reconhecer, diagnosticar e recuperar erros:

as mensagens de erro devem ser indicar em linguagem simples e com precisão o problema e sugerir construtivamente uma solução;

10. Ajuda e Documentação: mesmo que seja melhor se o sistema pode ser usado sem documentação, pode ser necessário fornecer ajuda e documentação.

Somente por meio de um projeto focado no *design* de interface e usabilidade que ambientes de aprendizagem serão construídos de tal forma que favoreçam o seu uso efetivo, priorizando assim os processos de ensino e aprendizagem.

2.2 AMBIENTE VIRTUAL DE APRENDIZAGEM

Foi criado um novo panorama educacional a partir da entrada das Tecnologias da Comunicação e Informação (TICs) que vem gerando novas experiências e ampliações metodológicas. Com isso, estas tecnologias vêm de forma significativa transformando a maneira de agir e refletir na educação.

O AVA é uma dessas tecnologias e está em constante uso no cenário educacional, utilizado por educadores e demais interessados no processo de ensino através do auxílio de tecnologias. Um AVA apresenta-se como um espaço virtual onde alunos e professores interagem sobre os conteúdos da disciplina, preparam trabalhos, se envolvem em discussões e aprendem em colaboração através de fóruns, *chats*, transferência de arquivos, entre outros (ROMERO et al., 2008) (DILLENBOURG, 2000).

“Com o advento tecnológico, os cursos de graduação e pós-graduação enxergaram uma nova modalidade de ensino. A Educação a Distância (EAD) atrelada às ferramentas tecnológicas proporciona, cada vez mais, acessibilidade ao ensino de qualidade. Assim, amplia-se a necessidade de Ambientes Virtuais de Aprendizagem (AVA), capazes de proporcionar aprendizagem e a interação dialógica.” (UFAL, 2011)(BITTENCOURT; BITTENCOURT; SANTOS, 2011).

Neste sentido, um AVA vai além de apenas disponibilizar um ambiente para o compartilhamento de conhecimento, permite comunicação com outros sujeitos,

gerando uma interação que permite a produção e disseminação do conhecimento por mais de um sujeito inserido no ambiente.

Sob o olhar tecnológico, um AVA agrega componentes técnicos para o seu desenvolvimento e funcionamento, como computadores, servidores, *softwares* entre outros.

Já as funcionalidades são definidas pelos seus requisitos. Conforme Gonzales (2005), as funcionalidades dos AVAs podem ser organizadas em quatro grupos de ferramentas, são elas: de Coordenação, de Comunicação, de Produção dos Alunos ou de Cooperação e de Administração, definidas a seguir:

- **Ferramentas de Coordenação** servem de suporte para organizar um curso e são utilizadas pelo professor para disponibilizar informações aos alunos;
- **Ferramentas de Comunicação** tem o objetivo de facilitar o processo de ensino-aprendizagem através da colaboração e interação entre os participantes, como fóruns de discussão e bate-papo;
- **Ferramentas de Produção e/ou Cooperação** dos alunos oferecem espaço de publicação e organização do trabalho dos alunos em grupo, através de perfil, portfólio, diário e mural;
- **Ferramentas de Administração**, que oferecem recursos de gerenciamento do curso.

Através destas ferramentas, o professor pode ter diversas informações de participação e progresso dos alunos no decorrer do curso.

Embora inúmeras funcionalidades apresentem-se promissoras para o processo de ensino e aprendizagem, alguns autores apresentam possíveis problemas decorrentes da utilização de AVAs. Valente et al. (2009) afirma que o *Moodle*, um dos AVAs mais utilizados e com inúmeras funcionalidades, tem se tornando apenas uma moda e não mais um fator de inovação uma vez que é escolhido por ser mais conhecido. Isto reforça o observado por Santana et al.

(2014), no qual aponta que o *Moodle* não é utilizado por completo com suas funcionalidades, além de apontar que muitas das funcionalidades não são de fato úteis para o processo de ensino e aprendizagem.

Segundo (IVO, 2015) por incluir tantas funcionalidades o *Moodle* deixa a desejar no aspecto funcional:

“Por padrão, ao se deparar com a página inicial do AVA, os usuários encontrariam uma lista enorme de cursos, dentre os quais deveriam identificar aqueles onde estão matriculados. Caso seus esforços de achar sua sala sejam frustrados há a opção de busca no rodapé da página. Se ainda assim você não conseguir encontrar o que quer, só lhe resta encontrar a tela de acesso (ou login) que redirecionaria à lista de cursos desejados. A profundidade de níveis entre cursos e disciplinas é confusa além da grande quantidade de informação desnecessária nas telas.”(IVO, 2015).

Assim, o usuário pode perder muito tempo tentando encontrar uma disciplina, isso não pode acontecer, o caminho tem que ser mais curto e objetivo possível. O autor supracitado ainda afirma que os problemas não ficam apenas em encontrar um curso, “os ícones são muitos, e minimamente autoexplicativos. Pessoas que não têm o hábito de uso ou nunca tiveram contato com a plataforma ficam perdidos e para concluir as ações dentro dela com sucesso necessitam repetir e repetir a mesma tarefa”.

Neste sentido, um AVA deve considerar muito mais sobre tecnologias educacionais do que inserir inúmeras funcionalidades ou apenas transferir as informações apresentadas presencialmente para o computador. Seu uso depende de um planejamento direcionado ao atendimento de objetivos educacionais, ou limitaremos o uso de modernos equipamentos para transmissão/reprodução passiva do ensino tradicional (SILVA, 2008). Este planejamento depende, em muito, da interface que será construída, na qual deve considerar a relação do usuário (estudante) com o ambiente (AVA).

Levando em consideração as citações de alguns autores supracitados acima, a interface é um ponto crucial para um AVA, pois é nela que o usuário vai interagir e realizar todas as suas atividades necessárias no decorrer do curso, deste modo, a interface se torna a principal peça de ligação do usuário ao ambiente virtual.

2.3 TRABALHOS RELACIONADOS

Foram encontrados alguns estudos que se correlacionam com o projeto que está sendo proposto, a seguir são citados três trabalhos e seus resultados.

Em um estudo de avaliação de sítios web, MACIEL et al. (2004) enfatizam que mesmo o método de avaliação heurística sendo um dos mais utilizados, não aprecia todos os critérios de avaliação, mas concentra-se em facilidade de aprendizado e flexibilidade.

Em outro estudo, SALLES et al. (2006) analisam a importância de uma interface em um software educativo e como um layout interfere na utilização e usabilidade, citam que criar uma interface amigável é uma tarefa complexa, pois é a partir dela que o usuário decidirá se continuará utilizando a aplicação ou não, e se conseguirá usufruir da maioria de suas funcionalidades.

Um outro trabalho publicado, SILVA (2011) verifica o processo de ensino-aprendizagem mediado por tecnologias que influenciam relações entre docentes e discentes, e considera que os ambientes virtuais devem garantir a interatividade, minimizando o sentimento de “solidão” dos alunos que estudam sozinhos, mas que participam das atividades realizadas virtualmente, ainda cita que os AVAs estimulam o trabalho de forma cooperativa, a autoria compartilhada e estimula a comunicação entre os envolvidos.

3 MATERIAIS E MÉTODO

Este capítulo apresenta as tecnologias utilizadas para o desenvolvimento do protótipo, como ferramentas de construção de protótipos, linguagens de programação *web* e o método para implantação.

Em Engenharia de *Software*, existem algumas formas de prototipação de *software*, os mais utilizados são protótipo de baixa fidelidade e de alta fidelidade, sendo o último o protótipo escolhido para este trabalho, conhecido por protótipo funcional, pois resulta num produto mais próximo da versão final.

Ressalta-se que o protótipo funcional desenvolvido trata-se de um *software web*, ou seja, é acessado pelo lado do cliente diretamente de um navegador de *internet* como *Mozilla Firefox*, *Google Chrome*, *Safari* entre outros. O acesso pode ser realizado através de um *Notebook* ou Computador, e também por um dispositivo Móvel como *Smartphone* ou *Tablet*, pois também é preparado para acesso via dispositivo móvel.

3.1 TECNOLOGIAS E FERRAMENTAS APLICADAS

O protótipo foi construído a partir de tecnologias e ferramentas específicas para o desenvolvimento de um *software web*. O Quadro 1 apresenta as tecnologias e ferramentas utilizadas para desenvolvimento do protótipo.

Nome	Versão	Site	Descrição
<i>Hypertext Preprocessor (PHP)</i>	7	https://php.net	Linguagem de Programação
<i>HyperText Markup Language (HTML)</i>	5	http://www.w3c.br	Linguagem de Marcação
<i>Cascading Style Sheets (CSS)</i>	3	http://www.w3c.br	Linguagem de Estilos
<i>Bootstrap</i>	3.3	http://getbootstrap.com	<i>Framework</i> HTML, CSS e JavaScript.
<i>Javascript (JS)</i>	1.8.5	http://www.w3c.br	Linguagem de Programação
<i>jQuery</i>	3.2.1	https://jquery.com	<i>Framework</i> Javascript
<i>Sublime Text</i>	3.0	https://www.sublimetext.com	Editor de Texto para Códigos
<i>EasyPHP</i>	17.0	http://www.easyphp.org	Ambiente de desenvolvimento com PHP, PHPMyAdmin, Apache & MySQL.
<i>Astah Community</i>	7.0	http://astah.net/editions/community	Ferramenta de <i>Design</i> de <i>Software</i>

Quadro 1: Quadro 1: Tecnologias e Ferramentas Utilizadas

Fonte: Elaboração do Autor (2017).

Por se tratar de um protótipo *web*, o Quadro 1 contempla em sua maioria tecnologias para o desenvolvimento *web*. As próximas seções abordam e detalham as linguagens de desenvolvimento *web* utilizadas, são elas: PHP, HTML, CSS e JS. Além da ferramenta *EasyPHP*.

3.1.1 Hypertext Preprocessor (PHP)

O PHP é uma linguagem de programação de ampla utilização, interpretada, que é especialmente interessante para desenvolvimento para a *web*. O objetivo principal da linguagem é permitir a desenvolvedores escreverem páginas que serão geradas dinamicamente rapidamente (PHP, 2017). A facilidade de uso da linguagem torna-a simples para um iniciante, mas oferece recursos para um programador

profissional, além de ser uma linguagem de código aberto, assim podendo ser adaptada para diferentes fins.

Segundo a *CodeCademy* (2017) o PHP pode: avaliar dados de formulários enviados por um navegador, criar conteúdo *web* personalizado para o navegador, conversar com um banco de dados, e até mesmo enviar e receber *cookies*.

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo *Linux*, várias variantes do *Unix* (como *HP-UX*, *Solaris* e *OpenBSD*), Microsoft Windows, Mac OS X, RISC OS e provavelmente outros. O PHP também é suportado pela maioria dos servidores *web* atualmente (PHP, 2017).

Um fator muito positivo a linguagem é que a grande maioria dos servidores de hospedagem utilizam PHP. Assim, o número de desenvolvedores vem aumentando consideravelmente. Estima-se que 20% do total de domínios da *Internet* usem a tecnologia atualmente.

O PHP é processado no servidor por isso é uma linguagem *Server-side*. Sendo assim suas aplicações não poderão ser copiadas por outras pessoas. Todos os processos, rotinas e funções serão feitas no servidor e o usuário receberá apenas o resultado em seu *browser* (APRENDER PHP, 2010).

Segundo o *website* oficial do PHP (2017), esses são os maiores campos onde os scripts PHP podem ser utilizados:

- **Scripts no lado do servidor (*server-side*).** Este é o mais tradicional e principal campo de atuação do PHP. Você precisa de três coisas para isto funcionar. O interpretador do PHP (CGI ou módulo do servidor), um servidor *web* e um navegador *web*. Você precisa rodar o servidor *web* conectado em uma instalação PHP. Você pode acessar os resultados de seu programa PHP com um navegador *web*, visualizando a página PHP através do servidor *web*. Tudo isso pode rodar na sua máquina pessoal se você estiver apenas experimentando programar com o PHP.

- **Scripts de linha de comando.** Você pode fazer um script PHP para executá-lo sem um servidor ou navegador. A única coisa necessária é o interpretador PHP.

Com o PHP, portanto, temos a liberdade de escolha de sistema operacional e de servidor *web*. Do mesmo modo, podemos escolher entre utilizar programação estruturada ou programação orientada a objeto (POO), ou ainda uma mistura das duas. (PHP, 2017).

3.1.2 Hypertext Markup Language (HTML)

O HTML, do português “Linguagem de Marcação de *Hyper*-texto”, é a principal linguagem utilizada na *web*. Ela permite a criação de documentos estruturados em títulos, parágrafos, listas, *links*, tabelas, formulários e em muitos outros elementos nos quais podem ser incorporadas imagens e objetos, como por exemplo, uma animação ou um vídeo (FLATSCHART, 2011).

A linguagem supõe facilidade para o desenvolvimento de *websites*, que é a interação com o PHP, ou vice-versa, que facilita muito para o programador a estruturar as suas páginas, mantendo a organização e qualidade do código, também pode interagir com JS, para adicionar mais interatividade com o usuário.

HTML é escrito por meio de *Tags*, que são delimitadas pelos sinais `<>` e `</>`, para identificação de conteúdo e função. Atualmente a HTML encontra-se na versão 5 e é padronizada pelo W3C (*World Wide Web Consortium*), uma organização internacional responsável por estabelecer padrões para a *internet*.

3.1.3 Cascading Style Sheets (CSS)

O CSS, do português “Folha de Estilo em Cascata” é uma linguagem de estilo, responsável pela formatação e apresentação de conteúdo: *layout*, cores, fontes entre outros (FLATSCHART, 2011).

Segundo o TABLELESS (2017) o CSS é uma linguagem utilizada para definir a apresentação (aparência) de documentos que adotam para o seu desenvolvimento linguagens de marcação (como XML, HTML e XHTML e etc.). O CSS define como serão exibidos os elementos contidos no código de um documento e sua maior vantagem é efetuar a separação entre o formato e o conteúdo de um documento.

Neste projeto é utilizado *Bootstrap*, um *framework* para desenvolvimento *web front-end* (HTML, CSS e JS) que adiciona às páginas recursos de responsividade, diversos componentes de interface e estilos leves e modernos. O *Bootstrap* é atualmente o principal *framework front-end* para desenvolvimento *web* responsivo. Além de um poderoso sistema de *grid*, que facilita a criação de páginas que se adequam aos mais diversos tamanhos de tela, o *Bootstrap* conta, ainda, com vários componentes de uso comum em aplicações *web*, como botões, menus, tabelas, etc. Tudo isso é alinhado a um conjunto de estilos visuais elegante e moderno, que reduz a necessidade de escrita de código CSS em boa parte das situações (DEV MEDIA, 2017).

3.1.4 Javascript (JS)

A linguagem *JavaScript* é uma linguagem de programação do tipo *Client Side*, ou seja, ela é executada no computador do usuário, com a função de adicionar mais interatividade. JS consegue interagir com, praticamente, todos os elementos de uma página HTML, trabalhar com variáveis, gerar resultados, alterar a aparência de elementos e sem a necessidade de ficar recarregando a página (SCHIMID, 2017).

Neste projeto foi utilizado *jQuery*, uma biblioteca de funções em JS que interage com o HTML, desenvolvida para simplificar os *scripts* interpretados no navegador do usuário (client-side). Criada em dezembro de 2006 no BarCamp de Nova York por John Resig, usada por cerca de 77% dos 10 mil sites mais visitados do mundo, *jQuery* é a mais popular das bibliotecas JavaScript. A *jQuery* foi

desenvolvida para ser uma biblioteca com suporte a qualquer navegador Web. Ela facilita a tarefa de desenvolvimento de aplicações em JS, tendo que atingir a enorme quantidade de navegadores em que nossa programação poderá ser executada (RICARDO, 2017).

3.1.5 Ambiente de desenvolvimento com EasyPHP Server

Quando é desenvolvido um *software*, primeiramente é hospedado de modo *offline*, ou seja, não fica disponível na *internet*, apenas para os envolvidos no projeto que são os programadores, *designers*, gerentes de projeto entre outros. Para isso, é necessário montar um ambiente de desenvolvimento em uma máquina local. Por isso, foi escolhido a utilização do *EasyPHP*, um *software* que permite tornar seu computador um servidor *Web Apache*, com a instalação de um módulo para programação em PHP, banco de dados MySQL, além de um módulo de administração de Banco de Dados chamado *PHPMysqlAdmin*. Com isso, temos o ambiente de desenvolvimento montado para iniciar o desenvolvimento do *software*.

Neste projeto foi utilizado a versão *DevServer*, segundo o *website* oficial do *EasyPHP* (2017), a versão *DevServer* instala um ambiente de desenvolvimento completo e pronto para uso. O *Devserver* é portátil, modular, totalmente configurável e fácil de atualizar e estender, também permite configurar um servidor local com as mesmas características do servidor de produção.

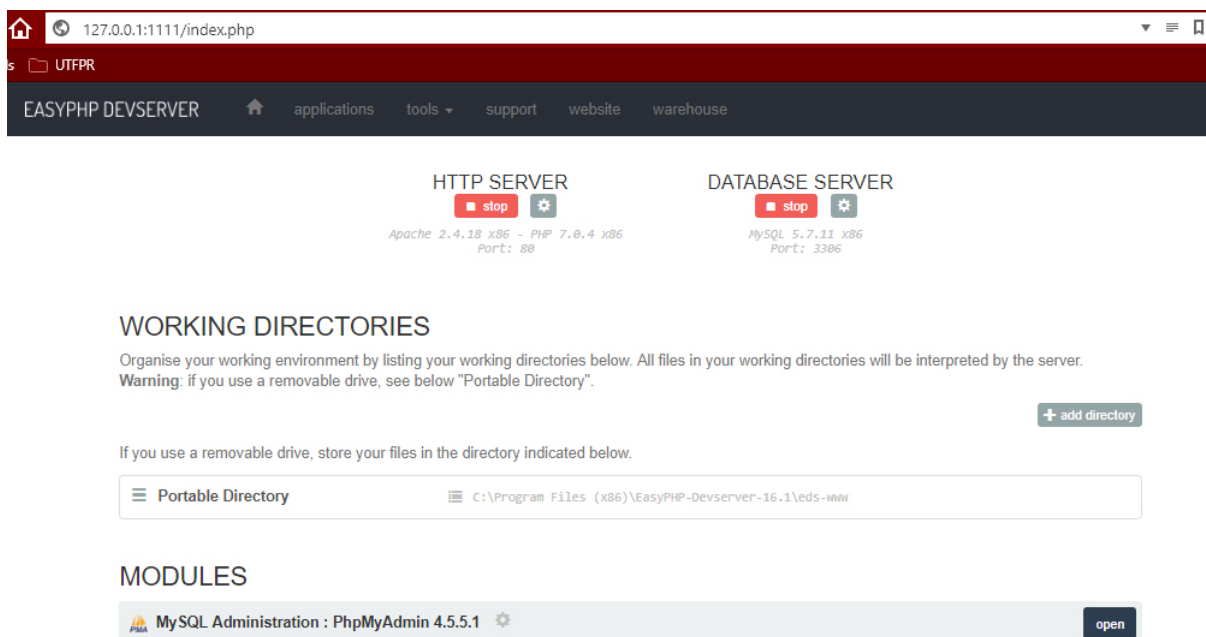


Figura 1: Tela inicial do EasyPHP

Fonte: Easy PHP (2017).

A Figura 1 exibe a página inicial da ferramenta, através de um navegador de *Internet* é possível acessá-la e configurá-la.

3.2 MÉTODO

O modelo escolhido para o desenvolvimento do protótipo, foi um modelo evolucionário de processo de *software*, denominado Prototipação de Pressman. Neste modelo, a construção do protótipo é essencial, e vem após o projeto de interface do *software*, onde possibilita experimentar opções de projeto, identificar problemas e criar um modelo do *software* que será implementado, além de facilitar a comunicação entre os desenvolvedores e os usuários. A Figura 2 apresenta o modelo de Prototipação utilizado.

O modelo é cíclico, fazendo com que cada fase seja realizada e podendo ser repetida para diferentes módulos. Desta forma, torna-se possível implementar os requisitos mais importantes e no decorrer do projeto se identificados novos requisitos, poderão ser implementados. Este modelo se caracteriza principalmente na criação de protótipos do sistema com as definições dadas pelo cliente. Esse

protótipo é então testado pelo cliente para validar suas funcionalidades (PRESSMAN, 2006).



Figura 2: Prototipação de Pressman

Fonte: Pressman (2011).

O paradigma da prototipação de Pressman (Figura 2) auxilia os *stakeholders*, ou seja, os interessados a entenderem melhor a forma como está sendo construído o *software*, que são as etapas:

- **Comunicação:** o desenvolvedor e o cliente encontram-se para definirem os objetivos gerais e levantamentos de requisitos do *software*;
- **Plano rápido:** esta etapa inicia-se o planejamento do protótipo com base na definição dos requisitos;
- **Modelagem rápida:** aqui a modelagem do protótipo inicia-se, a partir do *layout*, botões, cores etc;
- **Construção do Protótipo:** a programação do *software* é feita, porém, nem todas as funcionalidades são implementadas, pois é apenas uma versão para testes e validação;

- **Implantação, entrega e *feedback*:** a entrega do protótipo é realizada ao cliente para análise a espera de um *feedback* para ajustes.

A partir destas etapas, o desenvolvedor avalia se pretende evoluir o protótipo para chegar a versão final, ou iniciar o desenvolvimento a partir do zero até resultar na entrega do *software*.

Este modelo foi utilizado para construção do protótipo deste trabalho, sendo escolhido porque se adequa à construção de Ambientes Virtuais de Aprendizagem, apresentando rapidez na construção de protótipos e pela agilidade dos processos de comunicação e modelagem, importante quando há uma equipe heterogênea que precisa estar a par de cada etapa na construção do produto (PRESSMAN, 2006).

4 RESULTADOS

Este capítulo é destinado a apresentação do protótipo do *software* a partir de telas da interface, demonstrado as características e funcionalidades do *software* iniciando pelo escopo, em seguida a implementação que descreve como foi feita a construção, posteriormente a validação por meio das 10 Heurísticas de Nielsen e por fim a apresentação da interface.

O protótipo do *software* proposto foi nomeado de MyCODE pelo autor, assim, facilita o entendimento nas próximas Seções deste trabalho.

4.1 ESCOPO DO SOFTWARE

O MyCODE foi desenvolvido especialmente para apoio ao ensino presencial, pois possibilita que o aluno tenha uma opção especialmente fora da sala de aula para estudar a distância. Através dele, é possível realizar atividades devidamente propostas pelo professor, realizar *download* de arquivos como artigos, lista de atividades, livros em formato digital entre outros, interagir com outros alunos através de fóruns de discussão, etc.

Vale lembrar que este projeto objetivou a validação de interface do MyCODE, desta maneira o *software* possui funcionalidades limitadas por não ser versão final, ainda assim está devidamente planejada para trabalhos futuros.

Os requisitos foram definidos acompanhado do Professor orientador, analisando *softwares* similares encontrados atualmente como *CodeCademy*, *Khan Academy* e especialmente o *Moodle*. O resultado da análise de requisitos é mostrado através do Diagrama de Casos de Uso do Aluno (Apêndice A). Segundo (SOMMERVILLE, 2011) o diagrama de casos de uso é a técnica utilizada para a descoberta de requisitos, que dá nome e apresenta a interação entre o sistema e seus agentes externos, conhecidos como Atores.

4.2 IMPLEMENTAÇÃO DO PROTÓTIPO

O desenvolvimento do MyCODE foi dividido em duas etapas, a primeira é a construção do protótipo e a segunda a validação da interface por método definido no referencial teórico deste trabalho.

A primeira etapa da implementação foi a construção da interface do protótipo utilizando o *Framework Bootstrap* 3.3. Esta biblioteca facilita a construção de interfaces por meio de diversos componentes HTML, CSS e JS(*jQuery*).

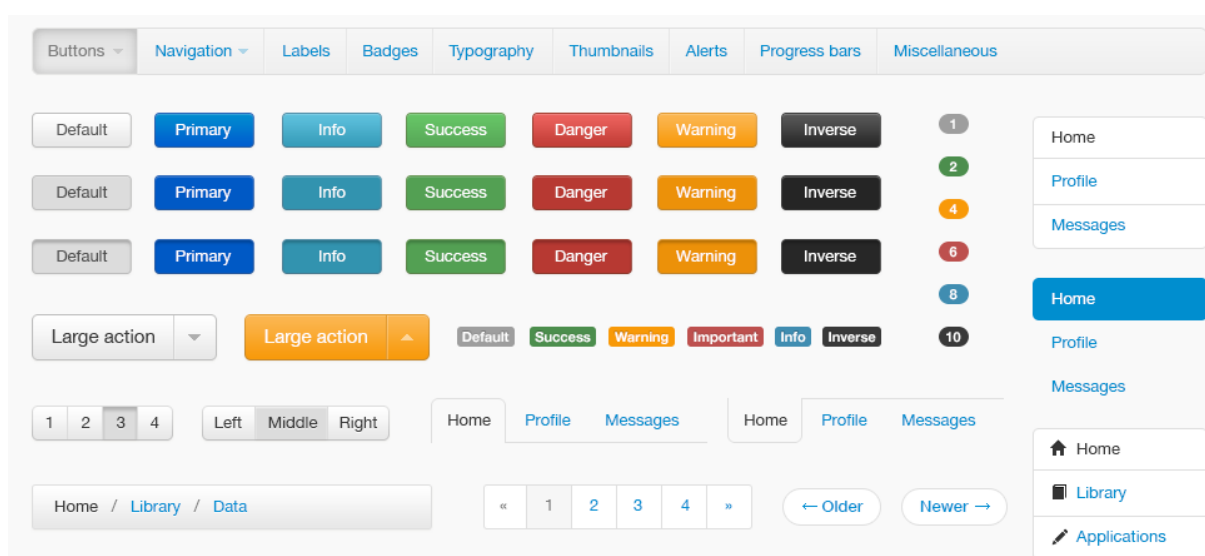


Figura 3: Diversos componentes do *Bootstrap*

Autor: Retirado da *Internet* (2017).

A Figura 3 mostra alguns componentes que o *Bootstrap* disponibiliza para uso, no site oficial do *framework* encontram-se todos os componentes disponíveis.:

Ainda assim, o *Bootstrap* não se limita apenas aos seus componentes, a principal vantagem de sua utilização é a responsividade das páginas através de um único código, ou seja, o *software* adapta-se a diferentes tamanhos de telas, resultando em um código funcional tanto para ambientes *desktops*, como para telas de *smartphones* e *tablets*. Desta maneira, seu uso torna-se essencial atualmente, quando há inúmeras plataformas pelas quais os alunos podem acessar o *software*.

A responsividade é proporcionada através do sistema de *Grids* que molda a interface de acordo com o dispositivo que o usuário está utilizando, conforme a Figura 4.

	Extra small devices Phones (<768px)	Small devices Tablets (≥768px)	Medium devices Desktops (≥992px)	Large devices Desktops (≥1200px)
Grid behavior	Horizontal at all times	Collapsed to start, horizontal above breakpoints		
Container width	None (auto)	750px	970px	1170px
Class prefix	.col-xs-	.col-sm-	.col-md-	.col-lg-

Figura 4: Sistema de Grids

Fonte: *Bootstrap 3.3* (2017).

Para o funcionamento dos componentes e sistema de *Grids*, é necessário o carregamento de um arquivo CSS e outro JS, além da biblioteca *jQuery* que é indispensável para o funcionamento das interações com o usuário. A Figura 5 demonstra o carregamento do arquivo “meta.php”, que faz a chamada dos arquivos por meio de código HTML embutido no arquivo PHP.

```

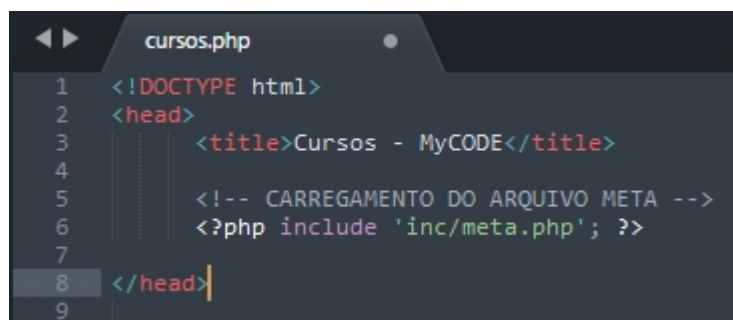
1 <!-- FAVICON -->
2 <link rel="shortcut icon" href="favicon.ico" type="image/x-icon">
3
4 <!-- TAGS -->
5 <meta charset="UTF-8">
6 <meta http-equiv="X-UA-Compatible" content="IE=edge">
7 <meta name="keywords" content="">
8 <meta name="description" content="">
9 <meta http-equiv="content-language" content="pt-br">
10 <meta name="robots" content="index, follow">
11 <meta name="reply-to" content="marcolin@alunos.utfpr.edu.br">
12 <meta name="viewport" content="width=device-width, initial-scale=1">
13 <meta name="author" content="Marcos Marcolin">
14
15 <!-- CSS -->
16 <link href="css/bootstrap.css" rel="stylesheet" /> <!-- CARREGAMENTO DO BOOSTRAP CSS -->
17 <link href="css/style.css" rel="stylesheet" />
18 <link href="css/dashboard.css" rel="stylesheet" />
19
20 <!-- JS -->
21 <script src="js/jquery-2.2.3.js"></script>
22 <script src="js/bootstrap.min.js"></script> <!-- CARREGAMENTO DO BOOSTRAP JS -->
23 <script src="js/funcoes.js"></script>
24 <script src="js/ckeditor/ckeditor.js"></script>
25
26 <!-- FONTES -->
27 <link href='https://fonts.googleapis.com/css?family=Open+Sans:400,700' rel='stylesheet' type='text/css'>
28

```

Figura 5: Carregamento do Bootstrap

Fonte: *Autoria Própria* (2017).

Desta maneira, é possível fazer a chamada uma única vez dos arquivos através de um código PHP não sendo necessário reescrever o código em todas as páginas, conforme a Figura 6.



```

1 <!DOCTYPE html>
2 <head>
3     <title>Cursos - MyCODE</title>
4
5     <!-- CARREGAMENTO DO ARQUIVO META -->
6     <?php include 'inc/meta.php'; ?>
7
8 </head>
9

```

Figura 6: Chamada do arquivo “meta.php”

Fonte: Autoria própria (2017).

Após a chamada dos arquivos, é possível utilizar todas as classes CSS e funções JS incluídas no *Bootstrap*, facilitando o desenvolvimento de interfaces responsivas.

Outra tecnologia utilizada no desenvolvimento do MyCODE é o *jQuery*, biblioteca JS que visa simplificar a escrita de código por parte do programador e facilitar a interação com o usuário. A Figura 7 apresenta a interação entre as linguagens HTML e CSS com *Bootstrap*, para realizar a interação de mostrar e esconder um formulário para criar um novo tópico no Fórum de discussões.



```

<h1 class="page-header">Fórum</h1>
<!-- FÓRUM -->
<!-- BTN-CRIAR TÓPICO -->
<button type="button" id="btn_mostrar_esconder_topico" class="btn btn-lg btn-success"><span class="glyphicon glyphicon-check"
aria-hidden="true" style="margin-right: 8px;"></span>CRIAR TÓPICO</button>
<br><br>
<!-- FORM NOVO TÓPICO -->
<form class="form-oculta" id="form_topico" style="display:none" method="post" action="processa_form.php">
  <div class="form-group">
    <label for="titulo">Título do Tópico:</label>
    <input type="text" class="form-control" name="titulo" required placeholder="Ex: Debate sobre a Atividade 5 de Programação">
  </div>
  <div class="form-group">
    <label for="conteudo_topico">Conteúdo do Tópico:</label>
    <textarea class="form-control" id="textarea_conteudo" name="conteudo_topico" rows="4" required placeholder="Ex: Dúvida no
conceito Switch Case"></textarea>
  </div>
  <input type="submit" class="btn btn-success" value="Criar" />
  <button type="reset" class="btn btn-default" id="btn_cancelar_topico" onclick="EsconderFormTopico">Cancelar</button>
</form>
<!-- FIM FORM NOVO TÓPICO -->

```

Figura 7: Código HTML e CSS do Formulário para criar um Tópico

Fonte: Autoria Própria (2017).

Complementando o código do formulário exposto na Figura 7, é chamada uma função *jQuery* para complementar mostrar e esconder o formulário conforme o clique em um botão do formulário, podendo ser para mostrar ou esconder o formulário, o código é apresentado na Figura 8.

```

1 // FUNÇÕES JS/jQuery
2 $(document).ready(function () {
3
4     // CRIAR TÓPICO
5     $("#btn_mostrar_esconder_topico").click(MostrarEsconderTopico);
6     function MostrarEsconderTopico(){
7         $("#form_topico").show(500);
8         $("#btn_mostrar_esconder_topico").hide(500);
9     }
10
11    // CANCELAR TÓPICO
12    $("#btn_cancelar_topico").click(EsconderFormTopico);
13    function EsconderFormTopico(){
14        $("#form_topico").hide(500);
15        $("#btn_mostrar_esconder_topico").show(500);
16    }
17
18 });

```

Figura 8: Código jQuery para mostrar e esconder elementos HTML

Fonte: Autoria Própria (2017).

A seguir, é possível observar os resultados dos códigos das Figuras 7 e 8 nas figuras 9 e 10. A Figura 9 expõe o ambiente Fórum antes de o usuário efetuar o clique no botão “CRIAR TÓPICO”. Aqui o formulário não fica visível através do código da Figura 7.

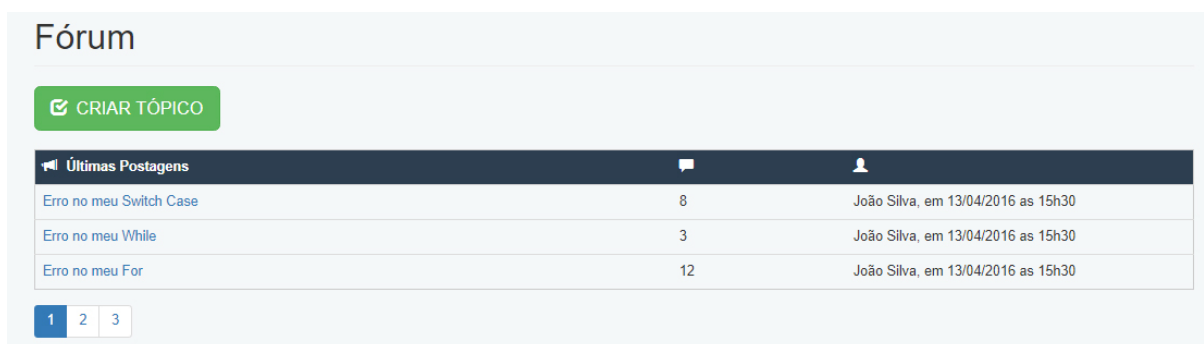


Figura 9: Ambiente do Fórum antes da interação do botão “CRIAR TÓPICO”

Fonte: Autoria Própria (2017).

Após o clique no Botão “CRIAR TÓPICO”, o formulário para inserção de um novo tópico fica visível (Figura 10). Para o usuário cancelar a criação do tópico e esconder novamente o formulário basta clicar no botão “Cancelar” que retornará a interface da Figura 9.

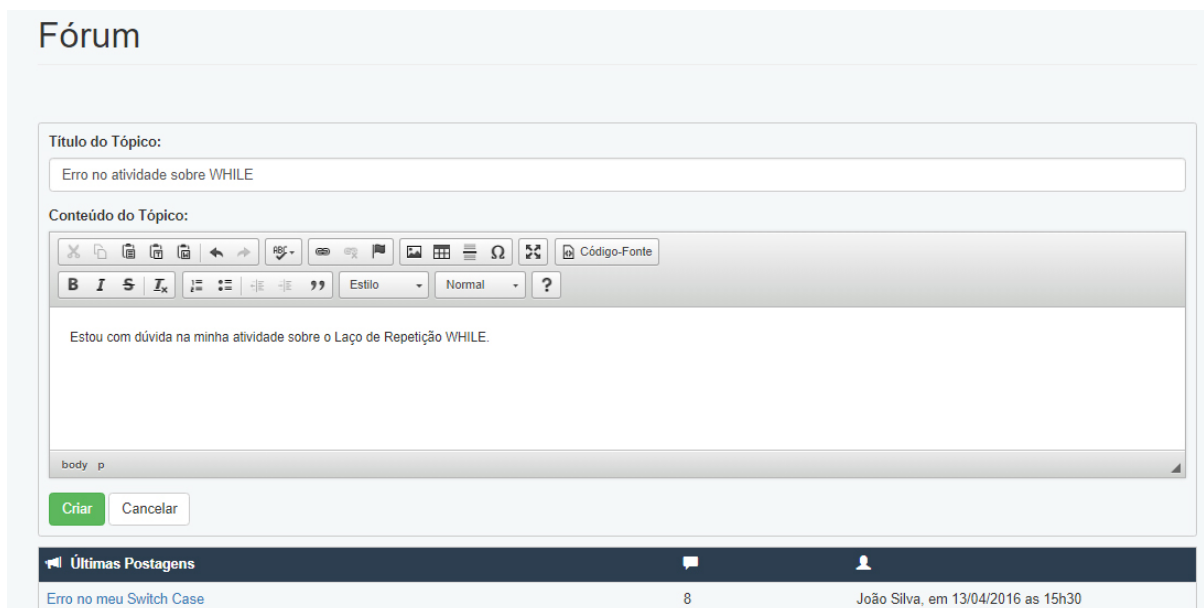


Figura 10: Ambiente do Fórum após a interação do botão “CRIAR TÓPICO”

Fonte: Autoria Própria (2017).

Após a construção da interface do MyCODE, sendo baseada nos princípios e nos exemplos oferecidos acima, a segunda etapa do desenvolvimento é iniciada, a validação da interface, sendo apresentada na próxima Seção deste trabalho.

4.3 VALIDAÇÃO DO PROTÓTIPO ATRAVÉS DAS 10 HEURÍSTICAS DE NIELSEN

Como já vimos na Seção 2.1 deste trabalho, as 10 Heurísticas de Nielsen são itens para avaliação de usabilidade de um site, com o intuito de evitar erros comuns por parte do usuário. Estas heurísticas foram baseadas em 294 erros de usabilidade encontradas em estudos pelo autor, e que podem seriamente prejudicar a experiência de um usuário ao acessar um site.

O objetivo desta seção é validar a interface do MyCODE através das Heurísticas, utilizando de telas comprovando que a interface do protótipo atende às exigências impostas por Nielsen.

4.3.1 Visibilidade do Estado do Sistema

Esta heurística requer que o sistema informe o que está acontecendo no *software*. No MyCODE, quando um aluno tenta criar um tópico no fórum, o sistema exibe um aviso da sua ação no momento, bem como avisa o usuário que os campos do formulário não podem ficar vazios conforme a Figura 11.

Você está criando um Novo Tópico para discussão, é obrigatório o preenchimento de todos os campos do formulário!

Título do Tópico:
Ex: Debate sobre a Atividade 5 de Programação

Conteúdo do Tópico:

Preencha este campo.

body p

Criar Cancelar

Figura 11: Criação de um Novo Tópico no Fórum
Fonte: Autoria Própria (2017).

4.3.2 Correspondência entre Sistema e Mundo Real

Nesta heurística é preciso considerar de qual forma o usuário se comunica com o sistema, neste contexto os ícones são fortes aliados para o usuário uma função que procura.

A Figura 12 exibe os ícones do menu superior e do menu responsivo, fazendo uma correspondência ao mundo real. Também o cursor do *mouse* for mantido sobre um ícone, o sistema exibe um pequeno *pop-up* com o título do ícone.



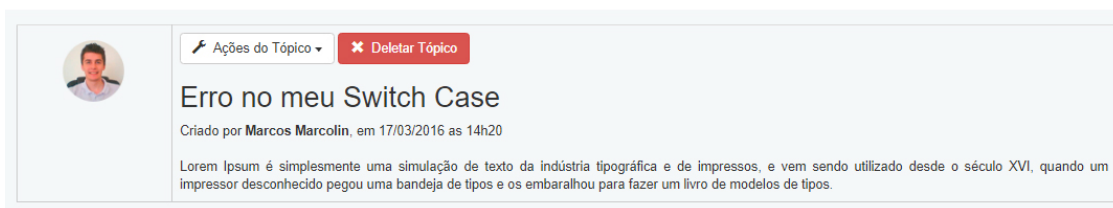
Figura 12: Correspondência dos Ícones do Menu do Aluno

Fonte: Autoria Própria (2017).

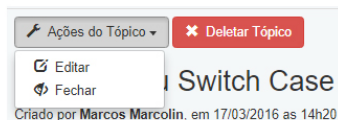
4.3.3 Controle e liberdade para o usuário

Esta heurística aponta que o usuário precisa do controle da situação, quando não está satisfeito com algo, ele pode desfazer-se a qualquer momento.

1. Botões visíveis para o Autor do Tópico



2. Interação Dropdown do Botão "Ações do Tópico"



3. Modo de Edição do Tópico

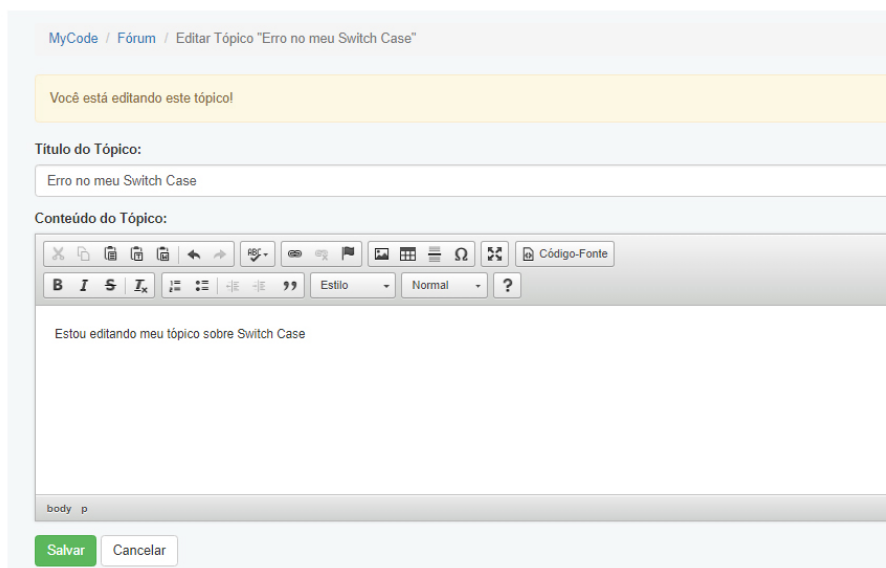


Figura 13: Ações para o autor de um Tópico

Fonte: Autoria Própria (2017).

No MyCODE isto é permitido, a Figura 13 demonstra um exemplo contendo três itens:

1. O primeiro item exibe os botões “Ações do Tópico” e “Deletar Tópico”, visíveis apenas para o Autor do Tópico, assim o autor tem controle total sobre o Tópico.
2. O segundo item mostra a interação com o botão *dropdown* exibindo as ações possíveis, sendo “editar” e “fechar”. Selecionando a opção “editar”, o autor é redirecionado a uma nova página, onde poderá realizar a edição do Título e do Conteúdo. A Opção “fechar” tem a função de bloquear o tópico, não podendo ser inserido novos comentários.
3. O terceiro item exibe o modo de edição do tópico selecionado pelo aluno. Após concluir a edição, o aluno pode Salvar, finalizando a edição.

4.3.4 Consistência e Padronização

Esta heurística aponta que o sistema precisa manter o padrão de botões, ícones, cores, fontes e tamanhos. A Figura 14 exibe o padrão de Botões e aviso, sempre facilitando ao usuário saber qual ação acontecerá. Por exemplo, a cor verde remete ao sucesso, a vermelha ao perigo e a amarela a um aviso.



Figura 14: Consistência de Padrões

Fonte: Autoria Própria (2017).

4.3.5 Prevenção de Erros

Por mais que o usuário precisa ter o controle de algumas situações, pode-se por um deslize clicar de forma involuntária em um botão que gere uma ação irreversível, por isso esta heurística aponta que deve-se prevenir os erros. A Figura 15 exemplifica utilizando o botão “Cancelar Matrícula”.



Figura 15: Prevenção de Erros

Fonte: Autoria Própria (2017).

4.3.6 Reconhecimento em vez de memorização

Esta heurística aponta que o usuário não precisa decorar todo o ambiente, mas sim o ambiente sinalizar bem a sua arquitetura de informações. Por exemplo, o sistema exibe em todas as páginas o caminho que o usuário percorreu até chegar em uma determinada página conforme a Figura 16, chamamos isso de *Breadcrumbs*.

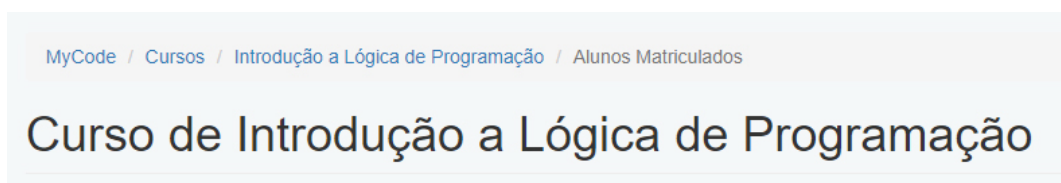
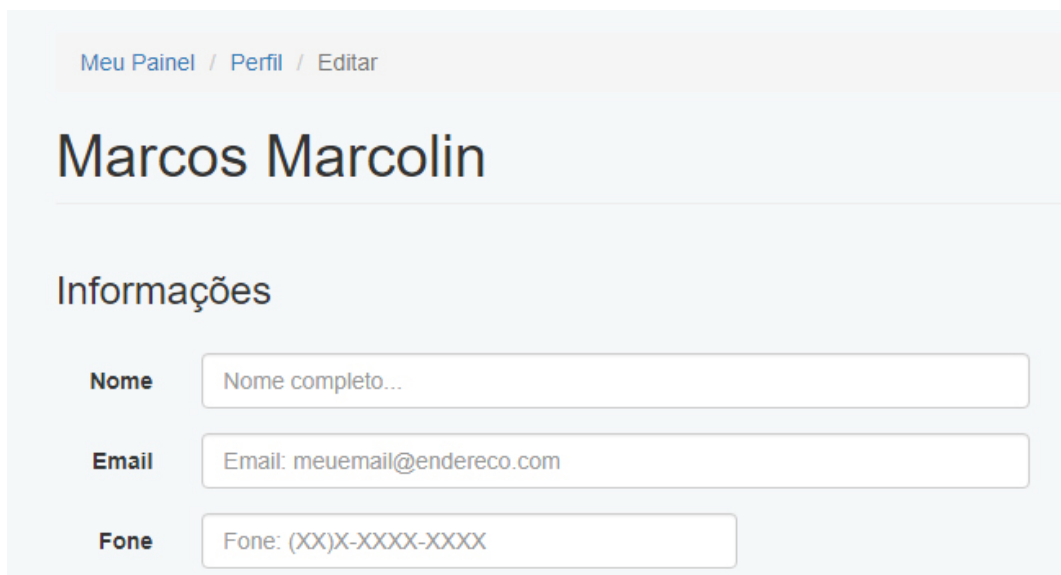


Figura 16: Breadcrumbs: Fluxo de Ações do Usuário

Fonte: Autoria Própria (2017).

4.3.7 Flexibilidade e eficiência de utilização

Esta heurística aponta que o sistema deve ser de fácil utilização tanto por um usuário leigo quanto para um avançado, desta forma deve-se facilitar seu uso mas também permitir rapidez de uso por usuários avançados. O preenchimento de formulário é um grande problema aos leigos, com a utilização de Máscaras de campo facilitamos o preenchimento das informações conforme a Figura 17. Para os usuários avançados, a sequência de informações e campos permite a utilização de teclas de atalho para navegação dentro do formulário.



Meu Painel / Perfil / Editar

Marcos Marcolin

Informações

Nome	<input type="text" value="Nome completo..."/>
Email	<input type="text" value="Email: meuemail@endereco.com"/>
Fone	<input type="text" value="Fone: (XX)X-XXXX-XXXX"/>

Figura 17: Máscaras de Campos em um Formulário

Fonte: Autoria Própria (2017).

4.3.8 Design estético e minimalista

O *layout* deve ser simples e objetivo, sem demais informações para fazer o usuário pensar muito para tomar uma decisão. No MyCODE o menu na versão *mobile* é um exemplo, onde encontra-se principais links do ambiente de acordo com a Figura 18.

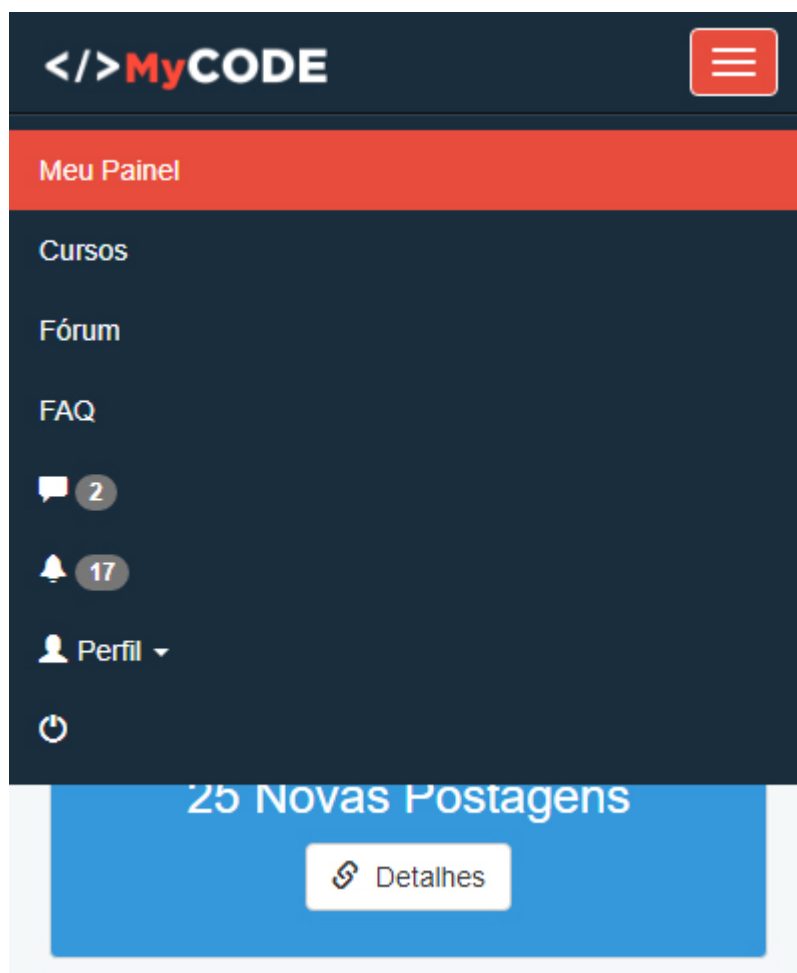
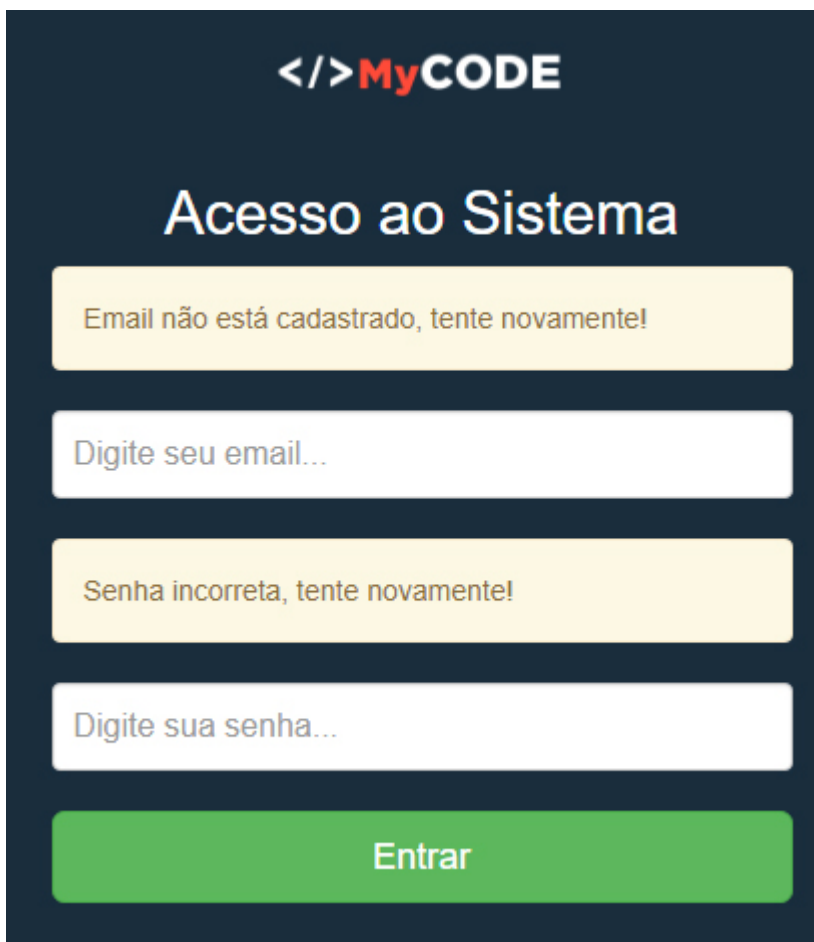


Figura 18: Design minimalista

Fonte: Autoria Própria (2017).

4.3.9 Ajude os usuários a reconhecer, diagnosticar e recuperar erros

Esta heurística aponta para o diagnóstico específico de erros para que o usuário saiba onde errou. No MyCODE, há controle de erros nos formulários e, como exemplo, ao realizar uma tentativa de Login; caso os dados estejam errados é informado ao usuário qual o problema ocorrido através de um aviso diretamente sobre o campo, conforme a Figura 19.



</>MyCODE

Acesso ao Sistema

Email não está cadastrado, tente novamente!

Digite seu email...

Senha incorreta, tente novamente!

Digite sua senha...

Entrar

Figura 19: Tela de Login

Fonte: Autoria Própria (2017).

4.3.10 Ajuda e Documentação

Esta heurística aponta para o suporte dos usuários por meio de ajuda e documentação. O MyCODE disponibiliza um FAQ com as perguntas mais frequentes dos usuários, onde pode ser acessado a qualquer momento para esclarecimento de dúvidas conforme a Figura 20.

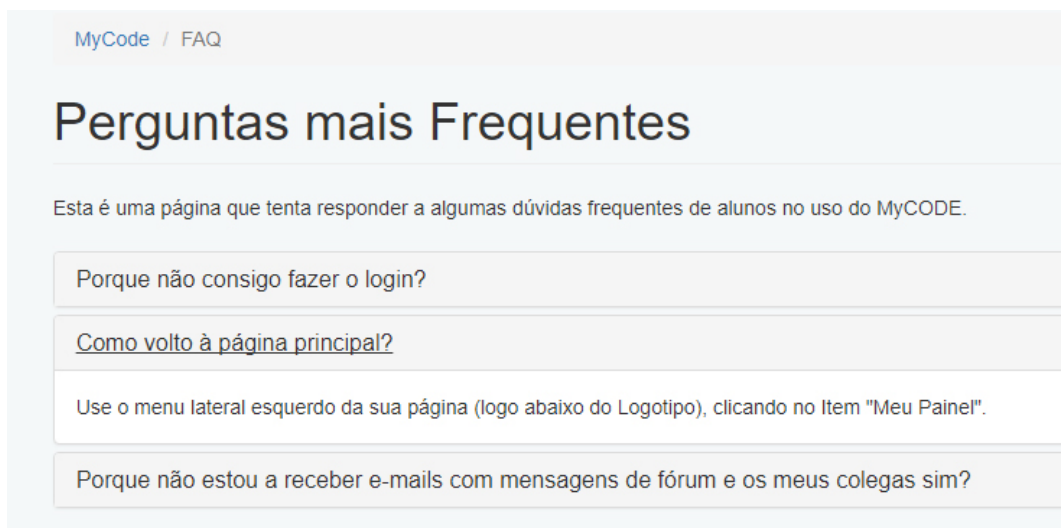


Figura 20: Perguntas mais Frequentes(FAQ)

Fonte: Aatoria Própria (2017).

Ao final, percebe-se que das Heurísticas ajudam na experiência do usuário para com o sistema, facilitando de forma significativa o acesso a todas as páginas, funções e ações.

Aplicar esta validação por meio das Heurísticas torna o sistema mais confiável perante o usuário, além de levantar problemáticas para os *stakeholders* nas interfaces desenvolvidas, assim prevenindo futuros problemas de usabilidade. Também, validar o sistema pode baratear o seu custo e reduzir o tempo de desenvolvimento.

4.4 APRESENTAÇÃO DO PROTÓTIPO

Nesta seção, são apresentadas as principais funcionalidades do MyCODE através de telas seguido de comentários e explicações do funcionamento de cada tela.

- **Meu Paine!:** nesta página o aluno tem a visão de algumas estatísticas do ambiente, bem como o menu lateral de acesso às demais páginas, além de um menu superior que mostras as notificações, mensagens e um menu *dropdown* com informações para o seu perfil.

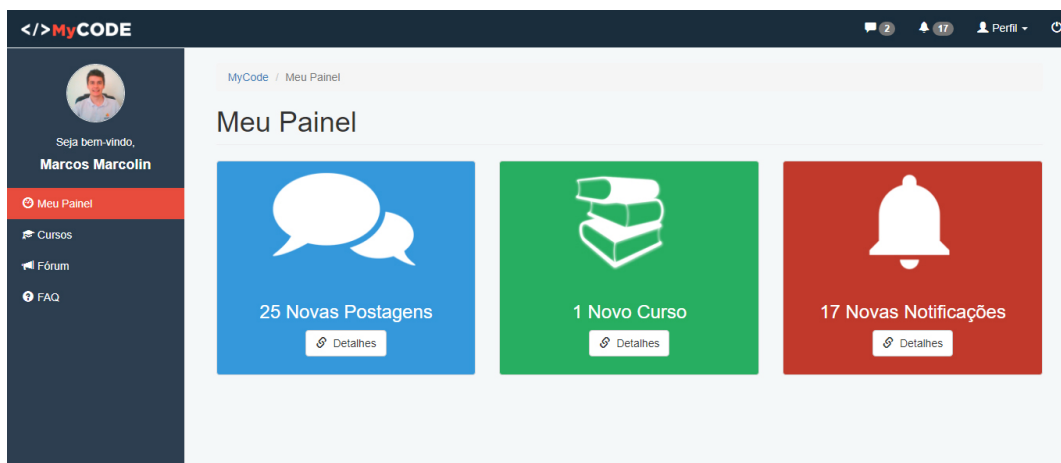


Figura 21: Interface Meu Painel

Fonte: Autoria Própria (2017)

- **Cursos:** nesta página o aluno tem acesso aos cursos disponíveis para fazer a inscrição, com o título do curso em destaque, a descrição, professor responsável e um botão que leva a página do curso.



Figura 22: Interface de Cursos

Fonte: Autoria Própria (2017).

- **Páginas do Curso:** após a inscrição, o aluno é direcionado a página do curso, onde ele recebe as boas-vindas do professor. Aqui ele tem acesso às atividades (Figura 23) que são divididas por módulos onde o aluno acompanha seu progresso, materiais oficiais da disciplina (Figura 24) e também aos alunos matriculados (Figura 25).

</>MyCODE 2 17 Perfil

Início Atividades **Materiais Oficiais da Disciplina** Alunos Matriculados

Seja bem-vindo,
Marcos Marcolin

Meu Painel

Cursos

Fórum

FAQ

Atividades!

Parte 1 - Introdução

Introdução a Linguagem C
Ao contrário do que se acredita, Lorem Ipsum não é simplesmente um texto randômico. Com mais de 2000 anos, suas raízes podem ser encontradas em uma obra de literatura latina clássica datada de 45 AC.
100%

Introdução a Linguagem C - 2
Ao contrário do que se acredita, Lorem Ipsum não é simplesmente um texto randômico. Com mais de 2000 anos, suas raízes podem ser encontradas em uma obra de literatura latina clássica datada de 45 AC.
50%

Progresso Total deste Módulo:
50%

Figura 23: Interface de Atividades

Fonte: Autoria Própria (2017).

</>MyCODE 2 17 Perfil

MyCode / Cursos / Introdução a Lógica de Programação / **Materiais Oficiais**

Curso de Introdução a Lógica de Programação

Início Atividades **Materiais Oficiais da Disciplina** Alunos Matriculados

Seja bem-vindo,
Marcos Marcolin

Meu Painel

Cursos

Fórum

FAQ

Materiais Oficiais disponibilizados pelo Professor!

Título		
Introdução À Programação - Uma Nova Abordagem Usando C	PDF	8
Algoritmos e Programação de Computadores	PDF	2
Introdução À Ciência da Computação - 2ª Ed. 2010	PDF	5
Introdução À Programação - Uma Nova Abordagem Usando C	PDF	8
Algoritmos e Programação de Computadores	PDF	2
Introdução À Ciência da Computação - 2ª Ed. 2010	PDF	5
Introdução À Programação - Uma Nova Abordagem Usando C	PDF	8
Algoritmos e Programação de Computadores	PDF	2
Introdução À Ciência da Computação - 2ª Ed. 2010	PDF	5

Figura 24: Interface de Materiais Oficiais da Disciplina

Fonte: Autoria Própria (2017).



Figura 25: Interface de Alunos Matriculados

Fonte: Autoria Própria (2017).

- **Fórum:** este ambiente é destinado a discussões iniciadas por alunos a fim de discutir atividades relacionadas ao curso, espera-se uma interação entre alunos e professores. Basta o aluno clicar no botão “CRIAR TÓPICO” para iniciar uma discussão. Após criá-lo, os seus colegas podem interagir conforme a Figura 27.

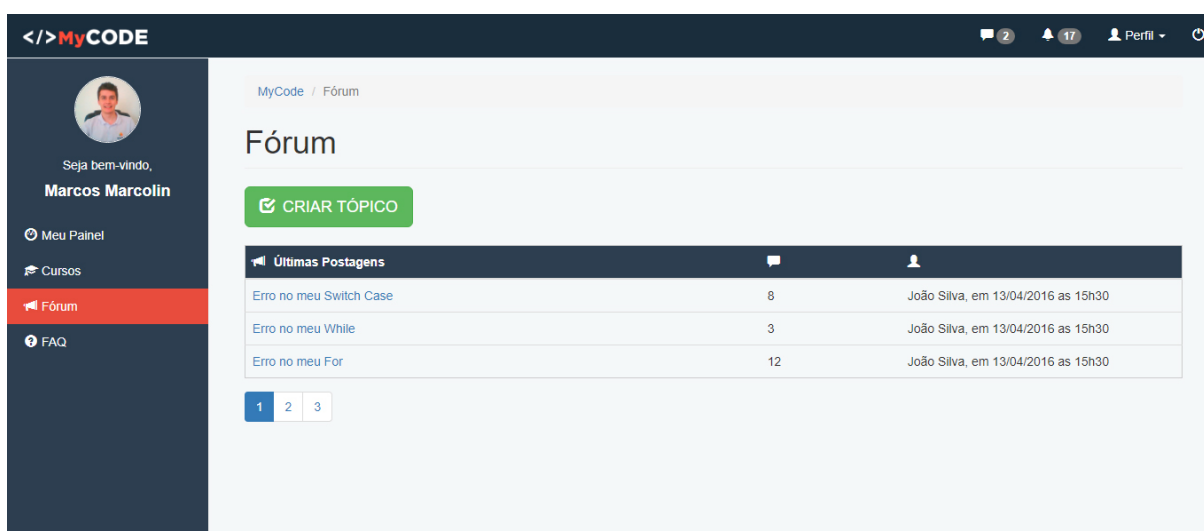


Figura 26: Interface do Fórum

Fonte: Autoria Própria (2017).



Figura 27: Interface de um Tópico no Fórum

Fonte: Autoria Própria (2017)

- **Perfil:** nesta parte o aluno tem acesso ao seu perfil, onde mostra seus dados cadastrados além de ter opções de editá-los quando achar necessário.

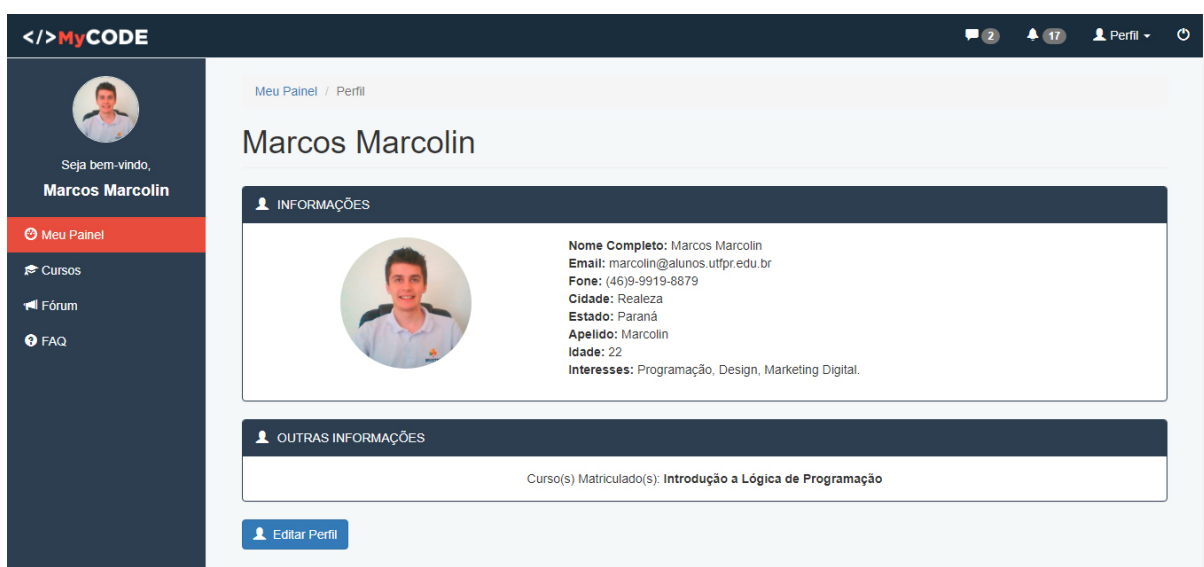


Figura 28: Interface Perfil do Aluno

Fonte: Autoria Própria (2017).

- **Interface responsiva:** neste item é apresentado algumas telas da interface responsiva para acesso em dispositivos móveis, como *tablets*

e *smartphones*. Todas as telas do MyCODE são responsivas, a Figura 29 exibe algumas delas.

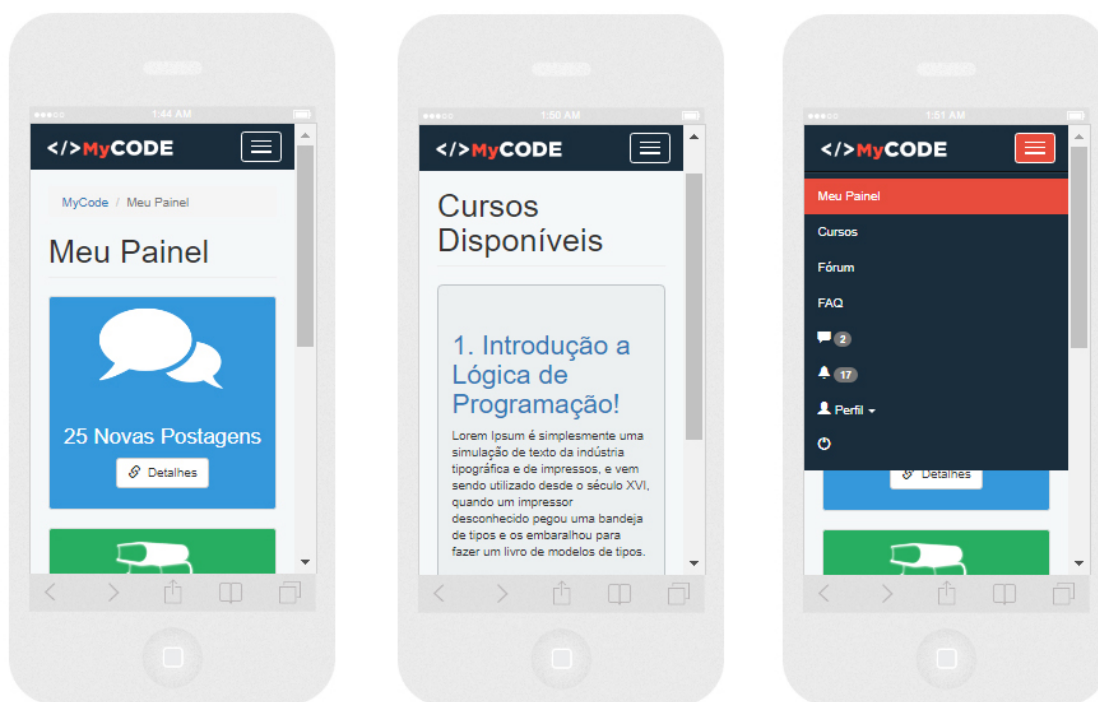


Figura 29: Interface de Ambientes Responsivos

Fonte: Figura gerada através da Ferramenta *Responsinator*, 2017.

5 CONSIDERAÇÕES FINAIS

Com a inserção das TICs nos ambientes educacionais, crescem as possibilidades de mudanças significativas nos métodos de ensino através do uso de novas tecnologias. Os Ambientes Virtuais de Aprendizagem (AVA) estão presentes neste contexto e fornecem suporte educacional especialmente no nível superior.

Durante a pesquisa do referencial teórico, foi constatado que com a demanda na construção de tais ambientes cresce também a importância do *design* de interface e usabilidade. Por muitas vezes estes conceitos são deixados de lado na criação de sistemas direcionados ao ensino, acarretando em falhas em sua implantação e por consequência, falha no cumprimento dos objetivos pedagógicos das ferramentas.

Ainda durante o levantamento teórico, ficou evidenciado que sistemas com interfaces mais limpas e objetivas, ou seja, com maior usabilidade, tornam-se mais efetivas nos seus resultados finais, além de tornar a experiência do usuário mais rica, gerando satisfação de uso, um elemento necessário para engajar o usuário na utilização de tais ambientes.

Este trabalho propôs o desenvolvimento de um protótipo funcional de *software web* para apoiar o ensino presencial, com foco na interface e na interação do usuário com o sistema. Para a construção do protótipo, foram utilizadas tecnologias voltadas ao desenvolvimento de aplicações *web*. As etapas da construção do protótipo foram determinadas pelo ciclo de vida utilizado, denominado Prototipação de Pressman, com este ciclo é possível identificar falhas na interface no processo de construção, podem ser corrigidas instantaneamente antes de lançar a versão final de um *software*. Em todo o processo de elaboração da interface, procurou-se sempre facilitar o caminho percorrido pelo usuário para acessar uma determinada página, assim, diminuindo o processo exaustivo de buscar por um determinado conteúdo, tornar a navegação segura evitando possíveis falhas que acarretam em erros irreversíveis dentro do ambiente e também utilizando uma

linguagem de comunicação de fácil entendimento para diversas situações fazendo com o que o usuário entenda o que está acontecendo em determinado momento. Após a construção da interface, foi realizado a validação por meio das Heurísticas de Nielsen (1994), foram criadas situações específicas para que atendam o que o autor propõe, assim, fazendo com que o *software* crie uma boa experiência de uso aos usuários.

Com o desenvolvimento deste trabalho, percebeu-se que o *design* de interface e usabilidade são fatores fundamentais para o desenvolvimento de um *software* nos dias atuais, onde a parte visual vem tornando-se cada vez mais importante. A importância do *design de interface* vem evoluindo com o tempo, desde as primeiras máquinas funcionais por terminais até chegar nos sistemas gráficos que temos hoje, apesar das funcionalidades serem essenciais em um computador ou *smartphone*, o *design* faz um grande diferença para o usuário, pois a experiência e facilidade de uso tornaram-se essenciais. Como resultado do protótipo, chegamos a um protótipo com *design* limpo, sem poluição visual, deixando uma visão clara ao usuário, com menus visíveis e de fácil acesso, consistência de padrões como botões, formulários e cores, aplicação totalmente acessível a dispositivos móveis e principalmente, interativa e confiável a ponto que o usuário consiga realizar suas atividades de maneira funcional.

5.1 TRABALHOS FUTUROS

Para trabalhos futuros, propõe-se o aperfeiçoamento ainda maior da interface, propõe-se também a validação por usuários finais, alunos regulares do ensino superior, também propõe-se a avaliação de usabilidade por meio de especialistas. Além disso, a implementação total da ferramenta, tendo as funcionalidades programadas, permitindo assim o uso efetivo da ferramenta.

6 REFERÊNCIAS

ALVES, L.; BARROS, D. M. V.; OKADA, A. **MOODLE: estratégias pedagógicas e estudos de caso**. [s.l.] Universidade do Estado da Bahia, 2009.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO/IEC 9126-1: **Engenharia de software - Qualidade de produto**. Rio de Janeiro, p. 9. 2003.

APRENDER PHP. **Introdução ao PHP**, 2010. Disponível em: <<http://aprenderphp.com.br/artigo/introducao-ao-php/>>. Acesso em: 30 de Set. 2017.

BENYON, David. **Interação humano-computador**. 2. ed. São Paulo, SP: Pearson Prentice Hall, 2011. 442 p.

BITTENCOURT, I. G. D. S.; BITTENCOURT, I. M.; SANTOS, C. N. **Usabilidade e os problemas do Moodle: o caso da Educação Universitária**, In: 17º Congresso Internacional ABED de Educação a Distância, Maceió, 2011. Anais de Educação a Distância. Maceió: Universidade Federal de Alagoas, 2011. Disponível em: <<http://www.abed.org.br/congresso2011/cd/260.pdf>>. Acesso em: 02 out. 2017.

CODECADEMY. **PHP em Ação**. Disponível em: <https://www.codecademy.com/pt-BR/courses/web-beginner-pt-BR-z94bw/0/1?curriculum_id=535968335d81967355000001>. Acesso em: 30 de set. 2017.

DEVMEDIA. **O QUE É BOOTSTRAP?**. Disponível em: <<http://www.devmedia.com.br/curso/o-que-e-bootstrap/1961>>. Acesso em: 30 de set. 2017.

FLATSCHART, Fábio. **HTML5: embarque imediato**. Rio de Janeiro, RJ: Brasport, 2011. xxi, 228 p. (Web conceitos & ferramentas).

IVO, Mariéllen. **A importância da usabilidade em AVAs**, 2015. Disponível em: <<http://www.ead.unimontes.br/nasala/importancia-da-usabilidade-em-avas/>>. Acesso em: 02 de Out. 2017.

RICARDO, José. **jQuery Tutorial**. Disponível em: <<http://www.devmedia.com.br/jquery-tutorial/27299>>. Acesso em: 30 de set. 2017.

KRAHE, E. D.; et al. **Desafios do trabalho docente: mudança ou repetição**. Revista Novas Tecnologias na Educação, v. 4, n.2, dez. 2006.

MACIEL, C., et al. **Avaliação heurística de sítios na web**. Escola de Informática do SBC-Centro Oeste(2004).

Magalhães, E., Gomes, V., Rodrigues, A., Santos, L., & Conte, T. (2010). **Impacto Da Usabilidade Na Educação a Distância: Um Estudo De Caso No Moodle IFAM**. Em Proceedings of the IX Symposium on Human Factors in Computing Systems (pp. 231–236). Porto Alegre, Brazil, Brazil: Brazilian Computer Society. Obtido de <http://dl.acm.org/citation.cfm?id=1999593.1999626>

MOZZAQUATRO, P. M.; MEDINA, R. D. **Avaliação do ambiente virtual de aprendizagem Moodle sob diferentes visões: aspectos a considerar**. In: Novas Tecnologias na Educação, v.6, n.2, dez. 2008.

NORMAN, D. A.; SPOHRER, J. C. **Learner-centered Education**. Commun. ACM , v. 39, n.4, p. 24–27, abr. 1996.

PHP. **Manual PHP**. Disponível em: <https://secure.php.net/manual/pt_BR/>. Acesso em: 30 de set. 2017.

PREECE, J. et al. **Human-Computer Interaction**, Addison-Wesley, 1994.

PRESSMAN, Roger S. **Engenharia de Software: Uma abordagem profissional**. 7. ed. Tradução: Ariovaldo Griesi e Mario M. Fecchio. Porto Alegre: AMGH Editora, 2011.

PRESSMAN, R.S. **Engenharia de Software**. 6ª Ed, McGraw-Hill, 2006.

SALLES, José Antonio Gameiro; COSTA, C. de A.; CARDOSO, Roberson Cotta. **Necessidades para o desenvolvimento de uma interface adequada para resultados de ensino-aprendizagem bem sucedidos**. Anais do, v. 4, 2006.

SANTOS, R., **Ergonomização das interação homem-computador**. Abordagem heurística para avaliação da usabilidade de interfaces. Rio de Janeiro: PUC Rio, 2000. (Dissertação de mestrado).

SILVA, Ivanda Maria Martins. **Interfaces digitais na Educação a Distância: Das salas de aula aos Ambientes Virtuais de Aprendizagem**. Revista Digital da CVA, 2011.

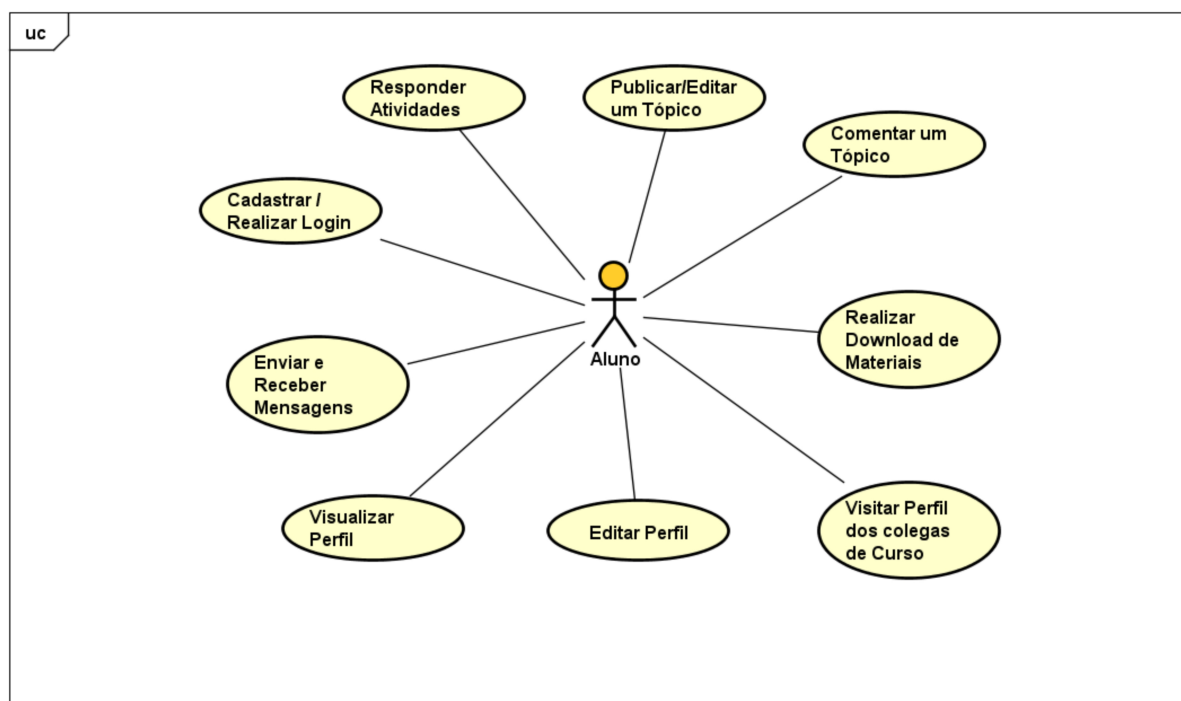
SHACKEL, B., Usability - context, framework, definition, design & evaluation, In: B. Shackel & S. J. Richardson, Eds., **Human Factors for Informatics Usability**, 21-37. Cambridge: Cambridge University Press, 1991.

SOMMERVILLE, I. **Engenharia de Software**. 9. ed. São Paulo: Pearson Prentice Hall, 2011.

TABLELESS. **Um guia para iniciantes na área de web..** Disponível em: <<http://tableless.github.io/iniciantes/>>. Acesso em: 30 de set. 2017.

7 APÊNDICES

APÊNDICE A – CASOS DE USO DO ALUNO



powered by Astah

Figura 30: Casos de Uso do Aluno

Fonte: Autoria Própria – Gerada pelo *Software Astah* (2017).