



UNIVERSIDADE FEDERAL RURAL DO SEMI-ÁRIDO
PRÓ-REITORIA DE GRADUAÇÃO
CENTRO MULTIDISCIPLINAR DE PAU DOS FERROS
BACHARELADO EM TECNOLOGIA DA INFORMAÇÃO

DYEGO MAGNO OLIVEIRA SOUZA

**DESENVOLVIMENTO E AVALIAÇÃO DO PROTÓTIPO DO *LOOP ACADEMIC*:
UM SOFTWARE EDUCACIONAL PARA O AUXÍLIO NO PROCESSO DE ENSINO-
APRENDIZAGEM DE PROGRAMAÇÃO INTRODUTÓRIA**

PAU DOS FERROS - RN

2019

DYEGO MAGNO OLIVEIRA SOUZA

**DESENVOLVIMENTO E AVALIAÇÃO DE UM PROTÓTIPO DE SOFTWARE
EDUCACIONAL PARA O AUXÍLIO NO PROCESSO DE ENSINO-
APRENDIZAGEM DE PROGRAMAÇÃO INTRODUTÓRIA**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal Rural do Semi-Árido -
UFERSA, *Campus* Pau dos Ferros, como
requisito parcial para a obtenção do grau de
Bacharel em Tecnologia da Informação.

Orientadora: Profa. Ma. Jarbele Cássia da Silva
Coutinho - UFERSA

PAU DOS FERROS - RN

2019

© Todos os direitos estão reservados a Universidade Federal Rural do Semi-Árido. O conteúdo desta obra é de inteira responsabilidade do (a) autor (a), sendo o mesmo, passível de sanções administrativas ou penais, caso sejam infringidas as leis que regulamentam a Propriedade Intelectual, respectivamente, Patentes: Lei nº 9.279/1996 e Direitos Autorais: Lei nº 9.610/1998. O conteúdo desta obra tomar-se-á de domínio público após a data de defesa e homologação da sua respectiva ata. A mesma poderá servir de base literária para novas pesquisas, desde que a obra e seu (a) respectivo (a) autor (a) sejam devidamente citados e mencionados os seus créditos bibliográficos.

S719d Souza, Dyego Magno Oliveira.
DESENVOLVIMENTO E AVALIAÇÃO DO PROTÓTIPO DO
LOOP ACADEMIC: UM SOFTWARE EDUCACIONAL PARA O
AUXÍLIO NO PROCESSO DE ENSINO-APRENDIZAGEM DE
PROGRAMAÇÃO INTRODUTÓRIA / Dyego Magno Oliveira
Souza. - 2019.
99 f. : il.

Orientadora: Jarbele Cássia.
Monografia (graduação) - Universidade Federal
Rural do Semi-árido, Curso de Tecnologia da
Informação, 2019.

1. Software Educacional. 2. Usabilidade. 3.
Prototipagem. 4. Teste de Usabilidade. I. Cássia,
Jarbele, orient. II. Título.

O serviço de Geração Automática de Ficha Catalográfica para Trabalhos de Conclusão de Curso (TCC's) foi desenvolvido pelo Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo (USP) e gentilmente cedido para o Sistema de Bibliotecas da Universidade Federal Rural do Semi-Árido (SISBI-UFERSA), sendo customizado pela Superintendência de Tecnologia da Informação e Comunicação (SUTIC) sob orientação dos bibliotecários da instituição para ser adaptado às necessidades dos alunos dos Cursos de Graduação e Programas de Pós-Graduação da Universidade.

DYEGO MAGNO OLIVEIRA SOUZA

**DESENVOLVIMENTO E AVALIAÇÃO DE UM PROTÓTIPO DE SOFTWARE
EDUCACIONAL PARA O AUXÍLIO NO PROCESSO DE ENSINO-
APRENDIZAGEM DE PROGRAMAÇÃO INTRODUTÓRIA**

Trabalho de Conclusão de Curso apresentado à
Universidade Federal Rural do Semi-Árido -
UFERSA, *Campus* Pau dos Ferros, como
requisito parcial para a obtenção do grau de
Bacharel em Tecnologia da Informação.

Defendido em: 19/03/2019.

BANCA EXAMINADORA


Jarbele Cássia da Silva Coutinho, Prof. Ma. (UFERSA)
Presidente


Laysa Mabel de Oliveira Fontes, Prof. Dra. (UFERSA)
Membro Examinador


Felipe Torres Leite, Prof. Me. (UFERSA)
Membro Examinador

À Aline Neel, Erick Killywiny e meu saudoso cachorro Victor, que estarão eternizados dentro das melhores lembranças que habitam em mim (In Memoriam).

Este trabalho é especialmente dedicado à minha mãe, Luciene Aparecida, ao querido Mauro Filho, e à grande orientadora Jarbele Cássia. Sem o vosso apoio, o caminho teria sido muito mais árduo.

AGRADECIMENTOS

Não poderia, de forma alguma, deixar este espaço em branco. Há muito a agradecer, e isso é o que farei aqui.

Início meus os agradecimentos direcionando-os a minha mãe Luciene Aparecida, que a todo instante se preocupou comigo, mantendo-me firme e forte durante os semestres do curso. Sou eternamente grato por essa mulher de alma pura, que com sorrisos e cuidados, contribuiu para tornar não somente este trabalho real, mas também o nosso sonho, que é concluir minha formação na área que desde pequeno sou apaixonado. Obrigado, mãe!

Prossigo com os agradecimentos agora direcionando-os a todos os parentes da minha tão grande família. Não citarei nomes, com o risco de esquecer alguém, entretanto, falo de forma abrangente, pois aqueles que se sentirem tocados com as minhas palavras saberão o quanto sou grato por todo o apoio. Sempre digo que tenho sorte por ter nascido na melhor e mais unida família que já presenciei. À todos vocês, obrigado!

Dando continuidade, dedico este parágrafo a minha orientadora Profa. Ma. Jarbele Cássia. Dentre todos os professores que tenho como inspiração, ela é uma das primeiras. A sua figura feminina transparece um aspecto maternal e a essência da sua alma é tão pura que torna prazerosos os momentos ao seu lado. Todas as lições que tive com ela, são e serão de grande valor para o profissional que venho me tornando. A paciência para aturar meus atrasos, a sabedoria para me guiar e educar e as palavras incentivadoras, com toda certeza, me fazem ver o quão grato sou por ter enfrentado este desafio ao lado dela. Ainda há muito a ser explorado, e assim faremos. À você, querida orientadora, obrigado!

Direciono meus agradecimentos ao querido e amado Mauro Filho. E este parágrafo é o quarto agradecimento da minha lista - só ele sabe o quanto sou supersticioso com números. Todas as lições de vida que tive ao seu lado moldaram o ser que sou hoje: um homem mais maduro, que aos poucos compreende a vida. Sem o seu apoio, este resultado talvez não tivesse sido alcançado, pois se eu estivesse sozinho nos momentos de turbulência e sobrecarga, talvez não estivesse aqui, neste momento, agradecendo por tudo. A sua essência é única e o seu dom de apoiar é especial. Serei eternamente grato por tudo. À você, Lobinho, obrigado! Ah, “sejamos fortes como um Lobo!”.

Dando continuidade, quero dedicar este parágrafo aos professores que, para mim, são inspiração. São profissionais únicos, que me fazem querer almejar ser tão grande assim, um dia. Todos os ensinamentos que tive ao seu lado me serão úteis na vida. Todas o reconhecimento e incentivo me fizeram prosseguir no curso dando o melhor de mim. E tê-los como inspiração é um sinal de que vocês são realmente incríveis! Aos Professores Jarbas Nunes (ausente), Marco Aurélio e Laysa Mabel, obrigado!

Dedico este espaço para agradecer aos grandes amigos que fiz no curso. Cada qual com sua personalidade única, que me faz ter a certeza de que boas pessoas existem no mundo. Vivemos diversos momentos juntos, e viveremos muitos outros, pois amizades assim são para toda uma vida. Déborah Assunção, Eliomar, D'angels, Raí Jácome, Rafaela (Rafinha), Kervin, Liliane, Mallu Barros, Brígido, Edgle, José Lucas, José Neto, Silvio, Lucas, obrigado!

Agradeço aos meus ex-alunos da monitoria de Algoritmos, que muito me ensinaram. Vocês fizeram crescer dentro de mim um Amor pela docência e me mostraram o quão prazeroso é dedicar todos os seus esforços em prol do outro. Obrigado pelos momentos, as notas máximas dedicadas e o grande laço de amizade que estabelecemos. Kennedy, Jefferson, Larissa Milena, João Victor, Joseano e Oscar, obrigado!

E por fim, deixo aqui meus agradecimentos aos companheiros de batalhas diárias. Chegamos à reta final do curso juntos e vivemos muitos momentos. Prosseguiremos na Engenharia de Software, mas mesmo assim, não posso deixar de agradecê-los por tudo! Josaias, Rarysson (Ford), Belarmino, Tássio, Assunauenny e Remyson, obrigado!

```
printf("Fim dos agradecimentos!");
```

“A única constante no mundo é a mudança”.
(Heraclito de Éfeso)

RESUMO

Este trabalho propõe o desenvolvimento do protótipo de um Software Educacional, denominado *Loop Academic*, direcionado ao auxílio no processo de ensino-aprendizagem de programação introdutória em cursos de nível superior, de modo a definir e realizar uma avaliação de usabilidade. A temática abordada é justificada pelos problemas enfrentados no processo de ensino-aprendizagem de programação introdutória em cursos de nível superior no Brasil, e de modo mais específico no curso de Bacharelado em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, campus Pau dos Ferros - RN. Assim, foi realizado um *survey* com alunos iniciantes na programação, com posterior validação dos requisitos, construção dos protótipos de baixa e de alta fidelidade, e aplicação de Testes de Usabilidade. Os resultados obtidos reforçam o potencial de uso do referido Software Educacional e direcionam para o desenvolvimento das funcionalidades apresentadas nos protótipos do Módulo Aluno do *Loop Academic*.

Palavras-chave: Software Educacional. Usabilidade. Prototipagem. Teste de Usabilidade.

ABSTRACT

This work proposes the development of the prototype of an Educational Software, denominated Loop Academic, aimed at assisting in the teaching-learning process of introductory programming in higher level courses, in order to define and perform a usability assessment. The thematic approach is justified by the problems faced in the teaching-learning process of introductory programming in higher education courses in Brazil, and more specifically in the Bachelor's degree in Information Technology at the Federal Rural University of the Semi-Arid, Campus Pau dos Ferros - RN. Thus, was carried a survey out with students beginners in the programming, with later validation of the requirements, construction of low and high fidelity prototypes and application of a Usability. The results obtained reinforced the potential of said Educational Software and directs to the development of the functionalities presented in the prototypes of the Student Module of the Academic Loop.

Keywords: Educational Software. Usability. Prototyping. Usability Test.

LISTA DE FIGURAS

Figura 01. <i>Card Sorting</i>	43
Figura 02. Escala SAM: Satisfação	47
Figura 03. Escala SAM: Controle	47
Figura 04. Escala SAM: Motivação	48
Figura 05. Diagrama de Casos de Uso do <i>Loop Academic</i> - módulo aluno	52
Figura 06. Persona 01: Tiago Include, o novato desamparado.....	53
Figura 07. Persona 02: Tereza Return, a veterana focada Fonte: Autoria própria	54
Figura 08. Logomarca do <i>Loop Academic</i>	55
Figura 09. <i>Favicon</i> do <i>Loop Academic</i>	55
Figura 10. Protótipo de baixa fidelidade: menu principal	56
Figura 11. Protótipo de baixa fidelidade: listas de exercícios	57
Figura 12. Protótipo de baixa fidelidade: resolução de exercício.....	58
Figura 13. Protótipo de baixa fidelidade: materiais de apoio	59
Figura 14. Protótipo de baixa fidelidade: estudo de um item do material de apoio	60
Figura 15. Protótipo de baixa fidelidade: desempenho	61
Figura 16. Protótipo de baixa fidelidade: dúvidas.....	62
Figura 17. Protótipo de baixa fidelidade: fórum	63
Figura 18. Protótipo de alta fidelidade: login	64
Figura 19. Protótipo de alta fidelidade: perfil.....	65
Figura 20. Protótipo de alta fidelidade: cadastro	65
Figura 21. Código da Turma Virtual	66
Figura 22. Protótipo de alta fidelidade: menu principal	67
Figura 23. Protótipo de alta fidelidade: listas de exercícios	68
Figura 24. Protótipo de alta fidelidade: exercícios de uma lista de exercício	68
Figura 25. Protótipo de alta fidelidade: exercícios de uma lista de exercícios.....	69
Figura 26. Protótipo de alta fidelidade: dicas de um exercício	70
Figura 27. Protótipo de alta fidelidade: simplificando o problema	70
Figura 28. Protótipo de alta fidelidade: descrição detalhada.....	71
Figura 29. Protótipo de alta fidelidade: materiais de apoio	72
Figura 30. Protótipo de alta fidelidade: itens de um material de apoio	72
Figura 31. Protótipo de alta fidelidade: itens de um material de apoio	73
Figura 32. Protótipo de alta fidelidade: excelente desempenho	74
Figura 33. Protótipo de alta fidelidade: baixo desempenho	75
Figura 34. Protótipo de alta fidelidade: dúvidas.....	76
Figura 35. Protótipo de alta fidelidade: dúvida respondida.....	76
Figura 36. Protótipo de alta fidelidade: envio de dúvidas	77
Figura 37. Protótipo de alta fidelidade: fórum	78
Figura 38. Protótipo de alta fidelidade: criar tópico no fórum	78
Figura 39. Protótipo de alta fidelidade: tópico do fórum	79
Figura 40. Protótipo de alta fidelidade: emblemas disponíveis.....	80
Figura 41. Protótipo de alta fidelidade: Emblemas Obtidos.....	81

LISTA DE QUADROS

Quadro 01. Heurísticas de usabilidade reformuladas por Benyon (2011).....	32
Quadro 02. Comparação entre trabalhos relacionados e o Loop Academic.....	39
Quadro 03. Questionamentos do survey aplicado com alunos.....	42
Quadro 04. Defeitos identificados no teste de usabilidade.....	82
Quadro 05. Tempos de execução das tarefas.....	83
Quadro 06. Intervalos de tempo para cada tarefa	84
Quadro 07. Opiniões dos participantes do teste de usabilidade	87

LISTA DE GRÁFICOS

Gráfico 01. Tipos de aluno de Algoritmos e Laboratório de Algoritmos no período 2018.2 ..	50
Gráfico 02. Utilidade do Loop Academic para o estudo de Algoritmos	50
Gráfico 03. Utilidade do Loop Academic para o incentivo aos estudos	51
Gráfico 04. Níveis de Acerto.....	82
Gráfico 05. Percentual de Defeitos.....	83
Gráfico 06. Resultados da Escala SAM: Satisfação	85
Gráfico 07. Resultados da Escala SAM: Controle	86
Gráfico 08. Resultados da Escala SAM: Motivação	86

LISTA DE ABREVIATURAS E SIGLAS

SE	Software Educacional
IHC	Interação Humano-Computador
UFERSA	Universidade Federal Rural do Semi-Árido
SAM	<i>Self-Assessment Manikin</i>
BTI	Bacharelado em Tecnologia da Informação
UFPB	Universidade Federal da Paraíba
GDSM	<i>Goal Driven Software Measurement</i>
UFAL	Universidade Federal do Alagoas
LABIE	Laboratório de Pesquisa em Informática na Educação
TCLE	Termo de Consentimento Livre e Esclarecido

SUMÁRIO

1 INTRODUÇÃO	16
1.1 CONTEXTUALIZAÇÃO	16
1.2 DELIMITAÇÃO DO TEMA E PROBLEMÁTICA	17
1.3 OBJETIVOS	17
1.3.1 Objetivo Geral	17
1.3.2 Objetivos Específicos	17
1.4 JUSTIFICATIVAS	18
1.5 ORGANIZAÇÃO DESTE TRABALHO	19
2 REFERENCIAL TEÓRICO	20
2.1 INFORMÁTICA NA EDUCAÇÃO	20
2.1.1 Ensino de Programação Introdutória	20
2.1.2 Software Educacional	22
2.2 <i>DESIGN</i> DE INTERAÇÃO	26
2.2.1 Elicitação de Requisitos	26
2.3 PERSONAS	29
2.4 PROTOTIPAÇÃO	30
2.5 AVALIAÇÃO DE INTERFACES	31
2.5.1 Inspeção de Usabilidade	33
2.5.2 Teste de Usabilidade	34
2.6 TRABALHOS RELACIONADOS	35
3 METODOLOGIA	41
3.1 PESQUISA DE OPINIÃO: UM <i>SURVEY</i>	41
3.2 PLANEJAMENTO PARA A PROTOTIPAÇÃO	43
3.2.1 Elicitação e Especificação dos Requisitos	43
3.2.2 Definição dos Personas	44
3.2.3 Prototipação de Baixa Fidelidade e de Alta Fidelidade	44
3.2.4 Definição das Ferramentas	44
3.3 PLANEJAMENTO E EXECUÇÃO DO TESTE DE USABILIDADE	46
4 ANÁLISE E DISCUSSÃO DOS RESULTADOS	49
4.1 RESULTADOS DO <i>SURVEY</i>	49
4.2 REQUISITOS FUNCIONAIS	51
4.3 PERSONAS	52
4.4 PROTÓTIPOS DESENVOLVIDOS	54

4.4.1 Componentes Iniciais	54
4.4.2 Telas do Protótipo de Baixa Fidelidade	55
4.4.3 Telas do Protótipo de Alta Fidelidade.....	64
4.5 RESULTADOS DO TESTE DE USABILIDADE	81
4.5.1 Níveis de Acerto.....	81
4.5.2 Tempo de Execução	83
4.5.3 Qualidade Afetiva	85
5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS	89
REFERÊNCIAS	91
ANEXO A	97

1 INTRODUÇÃO

A Introdução estrutura-se da seguinte forma: a Subseção 1.1 apresenta a contextualização acerca das principais temáticas relacionadas a este trabalho; a Subseção 1.2 demonstra a delimitação do tema e sua problemática; a Subseção 1.3 enfatiza os objetivos; a Subseção 1.4 demonstra as justificativas que culminaram na elaboração deste trabalho; e, por fim, a Subseção 1.5 apresenta a organização geral deste documento.

1.1 CONTEXTUALIZAÇÃO

Os computadores e seus recursos vêm sendo utilizados, de forma cada vez mais frequente, em todos os níveis da educação. Quando a informática é utilizada no apoio ao ensino, o aluno e o professor deixam de lado o quadro e o giz, considerado o método tradicional, e passam a alcançar diferentes patamares de ensino-aprendizagem (PAULA et al., 2014) (MORAES; WEBBER, 2017).

A integração da informática no processo de ensino-aprendizagem é uma questão bastante discutida nos contextos educacionais, pois tem ocasionado diversas mudanças em modelos educativos. Como resultados provenientes dessas mudanças, conteúdos complexos, numerosos e repletos de nuances de difícil compreensão puderam ser melhor trabalhados dentro e fora da sala de aula, o que culminou em um maior entendimento destes. Os resultados também abrangeram um aprendizado menos mecânico e desmotivador, demonstrando que aliar informática e educação foi um fator revolucionário às metodologias de aprendizagem (SILVA et al., 2017).

Um dos mecanismos utilizados para prover suporte ao processo de ensino-aprendizagem é o Software Educacional (SE) (ALMEIDA et al., 2018). Sua aplicação tem sido constante em diversas áreas, em especial, a área de programação introdutória em cursos de nível superior (CAFÉ et al., 2016).

De acordo com Pereira Júnior et al. (2005), Raposo e Dantas (2016) e Holanda, Coutinho e Fontes (2018), o ensino de programação introdutória têm enfrentado diversos problemas ao longo do tempo, que acabam por resultar em elevados índices de reprovação e desistência.

Assim, este trabalho de conclusão de curso propõe o desenvolvimento do protótipo de um software educacional denominado de *Loop Academic* direcionado ao auxílio no processo de ensino-aprendizagem de programação introdutória em cursos de nível superior, de modo a

definir e realizar uma avaliação de usabilidade, a fim de aferir o seu grau de usabilidade perante potenciais usuários.

1.2 DELIMITAÇÃO DO TEMA E PROBLEMÁTICA

Os índices de retenção em disciplinas de programação introdutória são elevados em boa parte dos cursos superiores de Computação no Brasil (SILVA et al., 2015; BRITO JÚNIOR; AGUIAR; TAVARES, 2016; QUEIROZ; RODRIGUES; COUTINHO, 2018). Partindo dessa afirmação e das dificuldades encontradas no ensino-aprendizagem de programação introdutória nas disciplinas iniciais de cursos superiores da área de Computação, esta proposta de pesquisa tem o intuito de responder a seguinte indagação: Como desenvolver um software educacional que considere os aspectos de usabilidade e auxilie no processo de ensino-aprendizagem de conteúdos de programação introdutória, ofertando aos alunos, professores e monitores, um ambiente capaz de auxiliar a resolução de atividades realizadas dentro e fora da sala de aula?

1.3 OBJETIVOS

Com base na problemática (descrita na Subseção 1.2), são apresentados os objetivos definidos para nortear o desenvolvimento desta pesquisa.

1.3.1 Objetivo Geral

O objetivo deste trabalho é desenvolver e avaliar o protótipo de um software educacional direcionado a auxiliar o processo de resolução de listas de exercícios de programação, utilizadas nas disciplinas de programação introdutória em cursos superiores da área de Computação, com o intuito de ofertar aos alunos: (i) materiais de estudo relacionados às suas principais dificuldades; (ii) dados representativos do seu desempenho; (iii) um ambiente propício à interação com outros usuários; (iv) incentivo aos estudos; (v) retirada de dúvidas.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral descrito na Subseção 1.3.1, definiu-se os seguintes objetivos específicos:

- I. Construir o protótipo de um software educacional seguindo algumas das diretrizes propostas pelo *Design* de Interação (ver Subseção 2.2);
- II. Investigar métodos avaliativos para avaliação de aspectos de usabilidade do protótipo desenvolvido;

III. Avaliar o protótipo desenvolvido.

1.4 JUSTIFICATIVAS

O ensino de programação em cursos de Computação e demais áreas que contam com o ensino de programação apresentam problemáticas em relação a aprendizagem do conteúdo abordado na sala de aula. Queiroz, Rodrigues e Coutinho (2018) apontam algumas estratégias que podem servir como apoio à aprendizagem de programação, a citar: (i) arduinos; (ii) ambientes de programação lúdicos; (iii) campeonatos de programação realizados na instituição de ensino; (iv) objetos de aprendizagem *online*; (v) jogos educativos; (vi) aplicativos de celular; (vii) grupos de estudos; (viii) computação desplugada. No que refere-se ao item ii, ambientes de programação lúdicos incluem SEs.

O projeto de um SE necessita de definições de requisitos que vão além do contexto imediato de uso. Decisões e seletividade sobre conteúdos, organização visual e didática dos elementos e adaptação perante os diferentes tipos de usuários são elementos necessários para o bom funcionamento de um SE (BASSANI et al., 2006).

Silva et al. (2016) ressaltam que um SE necessita satisfazer uma série de exigências que se encontram relacionadas: (i) à estruturação do conteúdo abordado; (ii) aos aspectos de aprendizagem; (iii) à coerência metodológica. Não obstante, é necessário que um SE também esteja de acordo com as premissas das Engenharias de Software e de Usabilidade, garantindo correteza funcional e adequação dos elementos de interação de sua interface, respectivamente; e que corresponda às Teorias Pedagógicas direcionadas a este tipo de software. A junção de todos esses fatores (funcionais, interacionais e pedagógicos) são decisivos na adoção de um SE.

No tocante aos fatores interacionais, as preocupações com a usabilidade em softwares educacionais são várias e estão relacionadas com a influência desses softwares na aprendizagem (NASCIMENTO et al., 2016). Por este motivo, é relevante submeter um SE a avaliações de usabilidade antes de sua aplicabilidade em contextos educacionais.

O autor deste trabalho também cita sua experiência como monitor da disciplina de Algoritmos na Universidade Federal Rural do Semi-Árido (UFERSA) - *Campus* Pau dos Ferros, nos períodos de 2017.1, 2017.2, 2018.1 e 2018.2. O contato mais próximo com os alunos permitiu o conhecimento de suas reais dificuldades e deficiências na questão do aprendizado da disciplina da área de programação introdutória. Esse contato foi o meio motivador para esta pesquisa, que se encontra fundamentada em possíveis soluções para as problemáticas enfrentadas pelos alunos iniciantes do curso de Bacharelado em Tecnologia da Informação – BTI da UFERSA, *Campus* Pau dos Ferros. Reforça-se que este trabalho também atua como

uma forma de melhor aproveitar todos os recursos didáticos que foram e estão sendo produzidos para o auxílio no processo de ensino-aprendizagem de Algoritmos, a citar: (i) as videoaulas produzidas pelo canal no *YouTube Programming Loop*¹; (ii) todo o material didático produzido pelo grupo de pesquisa LABIE (Laboratório de Pesquisa em Informática na Educação) da UFERSA, *Campus Pau dos Ferros*, o qual os autores compõem.

1.5 ORGANIZAÇÃO DESTE TRABALHO

O presente trabalho se desenvolve, estruturando-se, além desta Introdução, nas seguintes seções: a Seção 2 apresenta o Referencial Teórico, através de uma revisão da literatura relacionada à temática aqui estudada, bem como também apresenta os Trabalhos Relacionados com esta pesquisa; a Seção 3 discute a Metodologia, que demonstra todas as atividades desenvolvidas para o alcance do objetivo geral deste trabalho; a Seção 4 apresenta os Resultados Alcançados e Discussões, que aponta os resultados obtidos, enfatizando os protótipos construídos e as avaliações de usabilidade realizadas; e, por fim, a Seção 5 discute as Considerações Finais e os Trabalhos Futuros.

¹ Acesso ao canal: www.youtube.com/c/ProgrammingLoop

2 REFERENCIAL TEÓRICO

Esta seção apresenta o referencial teórico que contextualiza alguns conceitos gerais relevantes a este trabalho, estando organizada da seguinte forma: a Subseção 2.1 apresenta conceitos inerentes à informática na educação, com ênfase no ensino de programação introdutória, no cenário do ensino de programação em cursos de nível superior no Brasil, softwares educacionais, avaliação de SE; a Subseção 2.2 realiza uma revisão acerca do *Design* de Interação, com discussão também na Elicitação de Requisitos; a Subseção 2.3 fala sobre Personas; a Subseção 2.4 aborda os conceitos relacionados a Prototipação; a Subseção 2.5 aborda a Avaliação de Interfaces, teorizando Inspeção de Usabilidade e Teste de Usabilidade; e, por fim, a Subseção 2.6 apresenta os trabalhos relacionados a esta pesquisa.

2.1 INFORMÁTICA NA EDUCAÇÃO

O termo Informática na Educação refere-se a nova forma de procurar a formação do conhecimento, de modo mais autônomo e independente, em um novo espaço, onde os processos e as interações são diferentes (PAULA et al., 2014).

De acordo com Coutinho e Lisboa (2011), jovens alunos necessitam de novos modelos de ensino-aprendizagem para lhes garantir e estimular interatividade, proatividade, colaboração, inovação e outras diversas habilidades competentes e necessárias para o padrão exigido pela sociedade atual.

Nesse sentido, há várias possibilidades de uso das tecnologias nos cenários educacionais internos e externos às instituições de ensino. Independentemente do cenário de uso, o surgimento de ferramentas tecnológicas capazes de apoiar o processo de ensino-aprendizagem beneficiou a sua qualidade, o que também afetou diretamente métodos, práticas e recursos direcionados ao ensino. Muitas dessas ferramentas foram e são resultados do grande número de pesquisas na área da Informática na Educação, na qual várias questões de investigação continuamente tornam-se temas de estudos e posteriormente produtos direcionados à educação (COOMANS; LACERDA; PETESE, 2015; NASCIMENTO et al., 2016).

2.1.1 Ensino de Programação Introdutória

Dentre os conceitos fundamentais da área de computação, o de Algoritmos é um dos mais relevantes e necessários, pois embasa todo o conhecimento relacionado a esta área. Ao profissional da Computação, uma boa compreensão de Algoritmos permite o desenvolvimento de diversas outras competências necessárias ao seu perfil e área de atuação. Por este motivo,

cursos superiores em Computação ofertam disciplinas direcionadas ao estudo e implementação de algoritmos nos períodos iniciais (BARBOSA; FERREIRA; COSTA, 2014).

As instituições de ensino superior, preferivelmente, adotam linguagens de programação de sintaxe simples e de elevado potencial de abstração, pois possuem maior adequação ao ensino. Das diversas linguagens de programação em vigor no mercado de desenvolvimento de software, são comumente utilizadas nas instituições de ensino as linguagens: Pascal, C/C++, Java e Python (BARBOSA; FERREIRA; COSTA, 2014).

Em disciplinas de programação introdutória, são exigidas do aluno as seguintes habilidades e competências: (i) raciocínio lógico; (ii) resolução de problemas; (iii) capacidade de representar em uma linguagem de programação a abstração da resolução (JESUS; BRITO, 2009). O desenvolvimento dessas habilidades é possível através da prática de programação, que, por sua vez, somente é possível se o aluno compreender bem os conteúdos lecionados em sala de aula (CAFÉ et al, 2016).

Outro aspecto relevante na questão do ensino de programação introdutória está em seus conteúdos de caráter acumulativo. Tal característica leva alunos a se dedicarem aos estudos desde o início, a fim de terem capacidade de acompanhar a disciplina até o final (RAPOSO; DANTAS, 2016).

2.1.1.1 O cenário do ensino de programação introdutória em cursos de nível superior no Brasil

Para Pereira Júnior et al. (2005), o processo de ensino-aprendizagem de programação enfrenta diversos problemas, a citar: (i) dificuldade na abstração do exercício e interpretação de enunciados; (ii) dificuldade na identificação de pré-requisitos cruciais para o desenvolvimento dos algoritmos, dentre ao tais, habilidades matemáticas; (iii) dificuldade em aplicar habilidades prévias na construção de algoritmos. Raposo e Dantas (2016) elencam mais dois fatores que comprometem a qualidade do ensino de programação introdutória: (i) conteúdos de caráter acumulativo, que acabam por levar alunos que não se dedicam à incapacidade de acompanhar a disciplina até o final; (ii) a ausência de formas alternativas de estimular o ensino de programação e manter o aluno dedicado.

Contribuem para este cenário, a diversidade de ritmos de aprendizagem e a incapacidade do docente em ofertar atendimento individual contínuo, dado o quantitativo de alunos por turma (HOLANDA; COUTINHO; FONTES, 2018). Assim, a evasão nas disciplinas iniciais dos cursos de Computação, em especial nas disciplinas de programação, é fator de estudo para diversos autores nacionais e internacionais, que buscam encontrar soluções e/ou justificativas para esta problemática.

Segundo Giraffa e Mora (2013), a evasão ocorre no primeiro ano do ensino superior, o que abrange as disciplinas de programação introdutória, em especial a disciplina de Algoritmos.

Através de uma experiência, Campos (2010) apresenta dados em que mais de 60 % dos alunos matriculados em uma turma de Lógica de Programação foram reprovados por não terem as competências necessárias para a construção de algoritmos. Por sua vez, Raposo e Dantas (2016), apontam um cenário de três cursos da área de Computação da Universidade Federal da Paraíba - UFPB, cujos índices de aprovação na disciplina Introdução à Programação não ultrapassaram a margem de 34 % em nenhum dos seis últimos períodos anteriores ao ano de 2016.

Brito Júnior e Madeira (2015) afirmam que as disciplinas de introdução à programação há muito tempo apresentam altas taxas de reprovação, retenção e evasão, e enfatizam a necessidade de buscar formas de compreender melhor as causas reais dos problemas enfrentados pelos alunos ingressantes, através da criação de soluções que sejam capazes de remediar a atual situação de forma efetiva. Conforme Raposo e Dantas (2016), professores devem buscar formas alternativas de estimular o interesse dos alunos pelo conteúdo das disciplinas introdutórias de programação, mantendo-os em um ritmo constante de estudos.

2.1.2 Software Educacional

O uso de softwares educacionais no auxílio do processo de construção da aprendizagem tem se tornado uma estratégia bastante comum nos dias atuais, afirmam Almeida et al. (2018). A proposta do SE é dar suporte ao processo de ensino-aprendizagem (BASSANI et al., 2006) (TOLEDO, 2015).

De acordo com Silva et al (2016), um SE deve ser funcional, ou seja, seus elementos pedagógicos necessitam estar de acordo com o propósito dos conteúdos e conceitos abordados; sua estruturação deve garantir o correto funcionamento; e os elementos da sua interface devem condizer com as características e necessidades dos usuários. Qualquer deficiência evidenciada no SE pode prejudicar a construção de conhecimentos, influenciando negativamente no processo de ensino-aprendizagem.

Assim, um SE necessita estar pedagogicamente adequado às premissas estabelecidas pelo corpo docente, correto funcionalmente, conforme estabelece a Engenharia de Software e ser capaz de satisfazer características, necessidades e habilidades dos usuários, de acordo com a Engenharia de Usabilidade (DA SILVA; DE FRANÇA, 2014; SILVA et al., 2016).

De acordo com Jesus e Brito (2009), é válido ressaltar também que SEs devem estar desenvolvidos de tal forma a motivar a sua utilização pelos alunos e que estes encontrem nele

as ferramentas adequadas para auxílio no desenvolvimento das competências essenciais à construção do aprendizado.

Abreu et al. (2012) e Vieira (2007) citam que há diversas classificações para Software Educacional, baseadas em seus objetivos pedagógicos, e as tomam como primordiais para o projeto de desenvolvimento de um SE. Cada classificação permite guiar o desenvolvimento das características específicas a um tipo de SE.

A seguir, são citadas as classificações de softwares educacionais, de acordo com Abreu et al. (2012) e Vieira (2007):

- **Tutorial:** software educacional que transmite informações organizadas de forma pedagógica. Isto é, softwares educacionais com esta classificação disponibilizam a informação para o aluno seguindo uma sequência prévia, cuidadosamente definida e organizada pedagogicamente, de tal forma que a interação entre aluno e computador está fundamentalmente baseada na leitura da tela ou escuta da informação disponibilizada;
- **Exercícios e Práticas:** software educacional com enfoque em lições ou exercícios. Isto é, softwares educacionais com esta classificação apresentam exercícios e/ou atividades práticas ao aluno e mensuram resultados por intermédio da avaliação do computador;
- **Programação:** software educacional que permite ao aluno criar seus próprios protótipos de software, dispensando a necessidade de conhecimentos avançados em linguagens de programação. Softwares educacionais com esta classificação exigem que o aluno seja capaz de processar a informação e transformá-la em conhecimento. É importante enfatizar que: (i) o programa criado é reflexo das ideias do aluno; e (ii) existe uma correspondência direta entre os comandos utilizados e a forma como o computador irá comportar-se. No viés pedagógico, as características disponíveis no processo de programação auxiliam o professor a compreender a linha de raciocínio do aluno;
- **Multimídia e Internet:** software educacional que disponibiliza recursos multimidiáticos (textos, imagens, sons, etc.) para construção do conhecimento. É similar ao tutorial, pois as ações do aluno se restringem a escolher opções disponibilizadas pelo software educacional;
- **Simulação e Modelagem:** software educacional que simula situações e cenários difíceis, perigosos ou até impossíveis de serem reproduzidos em sala de aula.

Por exemplo: simular dissecação de cadáveres, realização de experimentos químicos, viagens na história, criação de planetas, etc. O processo de aprendizagem construído com este software educacional de simulação e modelagem exige atenção, pois é primordial propiciar ao aluno um ambiente com envolvimento do fenômeno descrito, e o direciona a levantar suas hipóteses, buscar outras fontes de informação e utilizar o computador para validar suas compreensões;

- **Jogos:** software educacional cuja finalidade é desafiar e motivar o aluno, de forma a envolvê-lo em uma competição, seja com o computador ou com outros jogadores.

2.1.2.1 Avaliação de Software Educacional

Avaliar um SE não é uma tarefa fácil. Se faz necessário identificar características satisfatórias quanto aos aspectos: (i) pedagógicos (qualidade do conteúdo); (i) de interface (usabilidade e experiência de uso); (iii) de elementos referentes à qualidade do software (aspectos de qualidade funcionais), conforme afirma Pereira et al. (2016).

Nesse sentido, várias pesquisas têm sido desenvolvidas na área, de modo a definir um padrão de avaliação mais comum ou mais condizente com as premissas de um bom SE. Assim, na tentativa de reduzir as dificuldades encontradas pelos profissionais da educação, várias abordagens de avaliação de SE têm sido propostas (SILVA et al., 2016), a citar: o Quali-EDU (LIMA et al., 2015); o guia *Goal Driven Software Measurement* - GDSM (Almeida et al., 2018); e o *framework* DECIDE (FANTIN, 2017).

O Quali-EDU é um processo de avaliação da qualidade de SE, elaborado com base na família de Normas ISO/IEC 9126 e na Norma ISO/IEC 14598. Seu principal objetivo é estabelecer um modelo de qualidade de SE que seja o mais próximo possível a professores e alunos (principais usuários de softwares dessa natureza), considerando aspectos de qualidade e educacionais (LIMA et al., 2015).

O guia *Goal Driven Software Measurement* – GDSM elabora indicadores para a avaliação de SE, com foco nos aspectos educacionais do processo de avaliação de software educacional Quali-EDU. No tocante à forma como ocorre a avaliação, os seus autores julgam relevante utilizar o método avaliativo do Quali-EDU e a Escala de *Likert* (ALMEIDA et al., 2018).

O *framework* DECIDE propõe e aplica uma metodologia de avaliação com base em diversos aspectos, tais como: (i) facilidade de aprendizagem; (ii) flexibilidade e eficiência; (iii)

prevenção de erros; (iv) ajuda e documentação; (v) satisfação; (vi) adequação pedagógica; (vii) utilidade e adequação técnica (FANTIN, 2017).

Apesar de algumas semelhanças entre os métodos avaliativos supracitados, nota-se que ainda não foi definida uma abordagem padronizada com completude igualitária dos três aspectos pedagógicos, de interface, e de qualidade de software. Para Pereira Júnior et al. (2005) existe uma diversidade de abordagens e configurações que abrangem a quantidade e natureza dos critérios avaliativos, tipos de instrumentos a serem utilizados na coleta de dados e o diagnóstico oferecido.

Assim, a necessidade de um SE satisfazer questões relacionadas à adequação pedagógica, corretude funcional e garantia de usabilidade é suficiente para considerar que, respectivamente, aspectos pedagógicos, de qualidade de software e de qualidade de uso devem ser tomados como principais fatores na avaliação de SE (PEREIRA JÚNIOR et al, 2005). Inadequações de qualquer natureza em um SE podem comprometer o seu propósito, reduzindo seu potencial ao auxílio no processo de ensino-aprendizagem (SILVA; FRANÇA, 2014).

Deste modo, dentre os principais aspectos considerados na avaliação de um SE, tem-se:

- **Avaliação pedagógica:** consiste em avaliar a didática e correspondência com teorias e conteúdos pedagógicos abordados nas disciplinas escolares, de forma a averiguar sua aptidão quanto instrumento de ensino-aprendizagem. Considerando os contextos estudantis, nos quais um software educacional tem aplicabilidade, é crucial que este possua qualidade pedagógica (FANTIN, 2017; SILVA, 2012);
- **Avaliação de usabilidade:** consiste em avaliar se o *design* do software é conciso, bem estruturado e de fácil comunicação e aprendizado. *Designs* complexos exigem uma maior curva de aprendizado do usuário. Para softwares educacionais, este fator tem um peso maior, pois com uma comunicação dificultosa com o usuário, o aprendizado é diminuído e gera-se uma insatisfação (BENYON, 2011);
- **Avaliação de qualidade:** consiste em avaliar aspectos intrínsecos a um software, tais como a funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade, portabilidade, eficácia, produtividade e satisfação (LIMA et al., 2015).

2.2 DESIGN DE INTERAÇÃO

A Interação Humano-Computador (IHC) almeja a garantia da usabilidade, enfatizando que os sistemas devem possuir facilidade de uso e aprendizado, flexibilidade e despertar no usuário boas atitudes (BENYON, 2011). Sendo assim, para compreender o significado de IHC, é necessário antes definir *Design* de Interação.

Conceitualmente, define-se *Design* de Interação como a forma de “projetar produtos interativos para apoiar o modo como as pessoas se comunicam e interagem” (ROGERS; SHARP; PREECE, 2013). Partindo deste conceito, enfatiza-se que a IHC não deixa, em nenhum aspecto, de ser *design* de interação, uma vez que também se preocupa com o *design*, a avaliação e a implementação de sistemas interativos para uso humano, porém, com o diferencial de restringir-se somente ao software, afirma Arnold (2011).

O tempo empregado para um usuário realizar determinada tarefa, o quantitativo de erros cometidos e os esforços despendidos para um usuário tornar-se competente são parâmetros que permitem mensurar a usabilidade de um software. Um sistema com alto grau de usabilidade será eficiente às pessoas, no sentido de que demandará menor esforço para cumprir atividades; será eficaz, fácil, seguro e útil (BENYON, 2011). Para o alcance da usabilidade, deverão ser considerados o *design* centrado no humano e uma abordagem avaliativa, continua Benyon (2011).

Para a construção de interfaces centradas nas premissas da IHC, é necessário compreender as atividades e aspirações dos indivíduos e os contextos nos quais uma determinada tecnologia tem utilidade. Diversas abordagens e técnicas adotadas pelas ciências sociais são utilizadas com esse intuito, a citar: a Sociologia, a Antropologia, a Psicologia e a Psicologia Cognitiva (ROGERS; SHARP; PREECE, 2013).

Benyon (2011) destaca que, na IHC, dois relacionamentos necessitam ser otimizados. A primeira relação é entre as pessoas e as tecnologias utilizadas, estando este relacionamento centrado na interface do usuário. A segunda relação é entre as pessoas e as tecnologias consideradas como um todo, estando esse relacionamento centrado nas atividades realizadas e em seus contextos, ou seja, na forma como o usuário faz uso da tecnologia.

2.2.1 Elicitação de Requisitos

A elicitação de requisitos consiste em uma das primeiras etapas realizadas no ciclo de vida da Engenharia de Requisitos. O seu objetivo é fornecer um mecanismo hábil para a

compreensão das necessidades do cliente, analisando a viabilidade destas e buscando uma solução válida e ausente de ambiguidades (PRESSMAN, 2016).

Assim, a elicitação de requisitos consiste, portanto e basicamente, na coleta e análise de requisitos. Esta é tida como a etapa inicial do ciclo de desenvolvimento de um software e necessita ser bem realizada, pois permite a documentação dos requisitos e construção dos demais artefatos que irão guiar o desenvolvimento do software (ENGHOLM JÚNIOR, 2010).

Em tese, a elicitação de requisitos é uma das atividades mais complexas de serem realizadas, pois é uma tarefa substancialmente humana que avalia diversos interessados e diferentes pontos de vista - o que já faz necessária uma análise minuciosa dos dados obtidos a fim de categorizá-los como requisitos úteis ao software a ser desenvolvido. Não obstante, há também o fator de impacto nas tarefas subsequentes do desenvolvimento de software (ENGHOLM JÚNIOR, 2010).

Sommerville (2011) resume o processo genérico de levantamento de requisitos nas seguintes atividades: (i) compreensão do domínio; (ii) coleta de requisitos, que consiste no processo de interação com *stakeholders* para descoberta de requisitos; (iii) classificação, que consiste em organizar o conjunto não estruturado dos requisitos e organizá-los em grupos coerentes; (iv) resolução de conflitos, que consiste em solucionar conflitos nos requisitos decorrentes dos requisitos elencados através de diversos *stakeholders*; (v) definição de prioridades, que consiste em definir prioridades aos requisitos, tendo em vista que alguns são mais relevantes do que outros devido às suas capacidades de melhor demonstrarem o escopo principal do sistema; (vi) verificação de requisitos, que consiste em analisar a completude, consistência e concordância de requisitos.

Destacam-se, ainda, a atividade de classificação de requisitos, pois é através desta que torna-se possível classificar os requisitos em: (i) requisitos funcionais, de acordo com aspectos funcionais; (ii) requisitos de interação, de acordo com aspectos de usabilidade.

Um requisito é compreendido como toda condição ou capacidade que deve ser implementada em um software, ou em sua parte, para que se torne possível o alcance de determinado objetivo (ENGHOLM JÚNIOR, 2010). Assim, o que torna um requisito válido é a sua necessidade, direta ou indireta, e a sua aprovação perante os *stakeholders* - o que assegura seu valor e alinhamento com os objetivos do projeto, segundo Engholm Júnior (2010).

Um requisito enquadra-se como funcional quando constitui as características e utilidades que os usuários do sistema esperam encontrar (ENGHOLM JÚNIOR, 2010). Para Sommerville (2011), um requisito funcional consiste nas definições de serviços ofertados pelo software, de reações quanto a entradas de dados específicas e de comportamentos perante

determinadas situações. Sommerville (2011) destaca ainda, que requisitos funcionais devem ser completos e consistentes. Completos quanto a abrangência de todos os serviços exigidos pelo cliente. Consistentes quanto ao fato de não existirem contradições e ambiguidades entre os requisitos. Para demonstração e melhor entendimento dos requisitos funcionais, utilizam-se Diagramas de Casos de Uso, definidos pela UML.

Um requisito de usabilidade é uma subcategoria dos requisitos não funcionais, que regem restrições sobre as formas nas quais os requisitos funcionais devem ser implementados. Um requisito não funcional é classificado como requisito de usabilidade quando define diretrizes para a implementação dos requisitos funcionais no que diz respeito à interface do software, e. g., maneiras que objetivam manter o usuário informado sobre o que ocorre no sistema (COSTA, 2018). O uso de protótipos (baixa, média e/ou alta fidelidade) apoiam o processo de definição e validação de requisitos de usabilidade (BARBOSA, 2008).

Para elicitar requisitos (funcionais e de usabilidade), existem algumas técnicas que auxiliam nesse processo. Sommerville (2011) e Engholm Júnior (2010) apontam as seguintes:

- **Levantamento Orientado a Pontos de Vista:** essa abordagem leva em consideração os diferentes tipos de usuários finais de um software, pois cada um tem um interesse nos requisitos ofertados pelo sistema. As abordagens utilizadas no Levantamento Orientado a Pontos de Vista objetivam reconhecer distintos pontos de vista e utilizá-los na fundamentação de requisitos. Para tal, três etapas possuem relevância: (i) identificar pontos de vista; (ii) estruturar pontos de vista identificados; (iii) documentação;
- **Etnografia:** essa abordagem tem como objetivo a compreensão de requisitos sociais e organizacionais do cenário no qual o software será/está inserido. Para tal, utilizam-se técnicas de observação, que consistem na inserção do analista no âmbito de aplicação do software para que, assim, seja possível descobrir requisitos de sistema implícitos, capazes de refletir processos reais;
- **Workshops:** essa abordagem consiste em uma técnica de elicitação em grupo, utilizada em uma reunião estruturada. O grupo é composto por analistas e *stakeholders* capazes de representar fielmente a organização e o âmbito no qual o sistema será aplicado, permitindo, de tal forma, a obtenção de um conjunto de requisitos bem definidos e elaborados. Difere-se da técnica de Reunião porque promove a interação entre todos os presentes;

- Prototipagem: essa abordagem consiste na exploração de aspectos críticos do sistema por meio da implementação de um subconjunto de funcionalidades do software. Através de protótipos, é possível analisar aspectos relacionados à interface e experiência do usuário, à viabilidade de atendimento dos requisitos de usabilidade e à compreensão dos requisitos elencados, com possibilidade de coleta de novos requisitos;
- Entrevistas: essa abordagem consiste na entrevista com *stakeholders*, estando o entrevistador aberto à exposição de ideias do entrevistado. Para sua boa execução, é necessário seguir um plano de entrevista, capaz de manter o foco nos aspectos representativos do software;
- Questionários: essa abordagem consiste na elaboração e aplicação de pesquisas específicas com usuários previamente selecionados e aptos a contribuir com informações relevantes que caracterizam requisitos para o sistema. As perguntas presentes em um questionário devem ser concisas, claras e simples, de forma a melhor guiar o entrevistado;
- Brainstorming: essa abordagem consiste em uma ou diversas reuniões que permitam e incentivem o surgimento e exploração de ideias. Participam analistas e clientes.

2.3 PERSONAS

Personas são tidas como sólidas representações dos diversos tipos de usuários de um sistema em desenvolvimento e devem possuir nome, alguns antecedentes, metas e aspirações (BENYON, 2011). Por meio de personas, é possível obter conhecimento sobre o público-alvo do sistema.

De acordo com Benyon (2011), é relevante desenvolver diferentes tipos de personas, pois permitem avaliar o sistema sob diferentes visões. Cada persona vai possuir suas metas e aspirações e se diferenciar dos demais nos níveis físico e psicológico e na forma como farão uso do sistema.

Pruitt e Grudin (2003) afirmam que personas aumentam o foco e a conscientização geral do usuário final em equipes de desenvolvimento através de cenários ficcionalizados. Este é um fator relevante, pois permite que o produto de software seja elaborado pensando no usuário e em suas possíveis interações. Segundo Guo, Shamdasani e Randall (2011), “ao definir o domínio no qual o trabalho de persona pretendido se aplica, podemos criar personas que afetam diretamente as decisões do produto relacionadas a esse espaço”.

Muito mais do que manter o foco do *design*, o uso de Personas resulta em uma base compartilhada, utilizada para comunicação entre os profissionais responsáveis pela interface e o público-alvo do software em desenvolvimento (PRUITT; GRUDIN, 2003). Assim, à medida que decisões de *design* são tomadas, consulta-se a base de personas para justificá-las de acordo com as características que foram definidas para cada persona do software.

2.4 PROTOTIPAÇÃO

O surgimento da Engenharia de Software levou à criação de muitos modelos de processos de software, propostos com o intuito de trazer ordem ao “caos” vigente na área de desenvolvimento e aumentar o quantitativo de projetos com sucesso (PRESSMAN, 2016). Dentre os modelos de processos de software, destaca-se aqui a Prototipação.

Para Bourque e Farley (2014), a prototipação é geralmente um meio para validar a interpretação do software para o engenheiro de software, bem como para suscitar novos requisitos.

O modelo de prototipação é recomendado nos casos em que: (i) o cliente possui uma ideia geral do software a ser desenvolvido e é incapaz de detalhar claramente os requisitos desejados; (ii) são ausentes detalhes de entrada, processamento e saída para as funcionalidades do software; (iii) houver insegurança quanto à forma de interação (PRESSMAN, 2016).

Este é um modelo preferível perante usuários e desenvolvedores, pois garante uma ideia prévia de como ficará o sistema, permitindo melhor captar requisitos e facilitando a codificação para a equipe de desenvolvimento. Outros aspectos vantajosos da prototipação estão na facilidade em realizar mudanças quando comparados a sistemas completos e consolidados; possibilidade de facilitar a interpretação e fornecer *feedback*; e contribuição para a redução de riscos de projeto (BOURQUE; FARLEY, 2014; PRESSMAN, 2016).

Para Bourley e Farley (2014), o comportamento dinâmico de uma interface de usuário é melhor compreendido na prototipação do que com a utilização de descrições textuais ou modelos gráficos, o que permite avaliar principalmente aspectos relacionados à IHC.

A prototipação é classificada em três categorias, de acordo com Rogers, Sharp e Preece (2013):

- **Protótipos de Baixa Fidelidade:** a prototipagem de baixa fidelidade não tem muita semelhança com o produto final, principalmente, devido ao fato de que são utilizados em sua concepção, materiais diferentes do material da versão final do produto - como papel, por exemplo. A utilidade de um protótipo de baixa fidelidade é justificada pela simplicidade, flexibilidade, baixo custo e menor

tempo de produção. Esses fatores também se aplicam às etapas de modificação do protótipo, o que encoraja fortemente a exploração de projetos e ideias alternativas, que ocorrem com frequência durante o projeto conceitual. Em suma, são protótipos rudimentares apenas destinados à exploração e nunca mantidos ou integrados ao produto final de software;

- Protótipos de Média Fidelidade: a prototipagem de média fidelidade é um meio-termo entre a prototipagem de baixa fidelidade e a prototipagem de alta fidelidade, pois agrega características de ambas. Em suma, são protótipos híbridos de implementação computadorizada de um software, funcionalmente limitado, e que permite a análise e avaliação de cenários específicos, não possuindo tanta abrangência e fidelidade quanto protótipos de alta fidelidade e não sendo tão rudimentar quanto protótipos de baixa fidelidade;
- Protótipos de Alta Fidelidade: a prototipagem de alta fidelidade é a mais próxima da versão final do produto de software. Em sua concepção, são utilizados materiais que o cliente espera encontrar no produto. Para tal, utilizam-se ambientes de desenvolvimento completos e capazes de suportar o processo de elaboração do protótipo de alta fidelidade. Também podem ser criados protótipos de alta fidelidade a partir de uma linguagem de programação. Em suma, são protótipos executáveis que contém as principais funcionalidades presentes na interface do produto final de software e levam em consideração aspectos estéticos (fonte, cor, tamanho de botões etc.) e componentes de navegação. Cabe enfatizar também que protótipos de alta fidelidade são utilizados no teste de usabilidade.

2.5 AVALIAÇÃO DE INTERFACES

A avaliação da IHC tem como intuito avaliar o grau de qualidade de uso da interface de um software. Com esse propósito, são utilizados métodos, técnicas e diretrizes que permitem aos desenvolvedores do software construir uma interface com o maior grau de qualidade possível (BARBOSA; SILVA, 2010). Quanto mais cedo for realizada a avaliação da qualidade da interface, menor será o custo do seu *redesign*.

De um modo geral, em IHC tem-se duas formas de avaliação: a avaliação de usabilidade e a avaliação centrada na experiência do usuário (do inglês *User Experience* - UX).

A avaliação de usabilidade é embasada pelos princípios da IHC. Conforme aponta Benyon (2011): “Os princípios de IHC podem orientar o designer durante o processo de *design*

e podem ser utilizados para avaliar e criticar ideias de protótipos”. Os princípios de IHC são compostos por uma série de heurísticas de usabilidade e têm se mostrado relevantemente úteis como instrumento de avaliação da qualidade de interfaces, principalmente na questão da usabilidade (COELHO; SANTORO, 2002).

As heurísticas de usabilidade, propostas por Nielsen (1994), constituem a base para a avaliação de usabilidade, e são categorizadas em: visibilidade do *status* do sistema; igualdade entre o sistema e o mundo real; controle e liberdade por parte do controle do usuário; consistência e padrões; prevenção de erro; flexibilidade e eficiência de uso; *design* estético e minimalista; facilidade dos usuários reconhecerem, diagnosticarem e se recuperarem de erros; ajuda e documentação.

Benyon (2011) afirma que os princípios de IHC, descritos por Nielsen (1994), possuem um alto nível de abstração, o que é justificado pela época na qual foram criados. Buscando maior adequação às tecnologias atuais, Benyon (2011) toma as heurísticas de usabilidade, definidas por Nielsen (1994) para criar uma lista de heurísticas de usabilidade reformulada.

O Quadro 01 apresenta as heurísticas de usabilidade reformuladas por Benyon (2011).

Quadro 01. Heurísticas de usabilidade reformuladas por Benyon (2011)

Nº	Heurística	Descrição
01	Visibilidade	Garantir visibilidade das funções e do estado atual do sistema. Caso não seja possível tornar visível, torne observável.
02	Consistência	Ser consistente no uso das características de <i>design</i> , com sistemas semelhantes e métodos-padrão de trabalho.
03	Familiaridade	Usar linguagem e símbolos familiares aos futuros usuários. Caso os conceitos sejam muito diferentes daqueles que os usuários conhecem, fornecer metáforas que os auxiliem a transferir conhecimentos similares e correlatos de um domínio mais familiar.
04	<i>Affordance</i>	Criar o <i>design</i> das coisas de forma que fique evidente o seu propósito e sejam respeitadas culturas de uso.
05	Navegação	Proporcionar suporte para que os futuros usuários sejam capazes de navegar pelo software.
06	Controle	Deixar claro quem ou o que está no controle e permitir que os futuros usuários também possam estar no controle.
07	Retorno (<i>feedback</i>)	Retornar rapidamente a informação do sistema para os futuros usuários, permitindo que saibam quais efeitos suas ações causaram.

08	Recuperação	Tornar possível recuperar-se de forma rápida e eficaz de ações, em especial enganos e erros.
09	Restrição	Restringir operações perigosas mediante a confirmação ou impossibilidade de realizá-las.
10	Flexibilidade	Proporcionar múltiplas maneiras de executar as funcionalidades do software, de forma a atender usuários com diferentes graus de experiência e interesse pelo sistema.
11	Estilo	Utilizar <i>designs</i> elegantes e atraentes.
12	Sociabilidade	Prezar pela boa educação, principalmente na forma como as mensagens são direcionadas ao usuário.

Fonte: Adaptado de Benyon, 2011

Ainda em relação à avaliação de usabilidade, os métodos comumente adotados encontram-se divididos em duas categorias: inspeções de usabilidade e testes de usabilidade (MATERA et al., 2002). As subseções a seguir descrevem, com maior viés, os dois métodos empregados na avaliação de usabilidade.

2.5.1 Inspeção de Usabilidade

A inspeção de usabilidade é uma avaliação na qual é dispensada a participação do usuário final, pois inspetores com conhecimento na IHC se encarregam de examinar aspectos da aplicação, com o intuito de encontrar violações na interface (VALENTIM, 2017) (MATERA et al, 2002). A partir das violações identificadas, são realizadas melhorias visando a eliminação dos aspectos que prejudicam a usabilidade. É válido ressaltar que a inspeção de usabilidade pode ser realizada em qualquer fase do desenvolvimento do software, dada a condição prévia de haver necessidade de avaliar a interface (ROCHA; BARANAUKAS, 2003).

De acordo com Valentim (2017), a inspeção de usabilidade pode ser realizada através de diferentes métodos, a citar:

- Avaliação Heurística: envolve um seletivo grupo de avaliadores destinados a examinar a interface e avaliar suas características de acordo com as heurísticas de usabilidade;
- Walkthrough Cognitivo: consiste em avaliar a interface quanto a facilidade de aprendizagem. Neste método, um aspecto da interface é apresentado à um grupo de pares, que ficam encarregados de avaliar a solução considerando critérios adequados ao *design* específico;

- Guidelines e Checklist: fundamentado em um conjunto de recomendações ergonômicas que auxiliam o processo de construção da interface, dos quais devem ser tomados como base. Posteriormente, é realizada uma inspeção minuciosa, que objetiva avaliar se todas as recomendações foram seguidas.

2.5.2 Teste de Usabilidade

O teste de usabilidade é uma avaliação realizada com a participação dos usuários finais, analisando suas interações com o software (ROCHA; BARANAUKAS, 2003). Em tese, é uma avaliação centrada no usuário. A sua aplicação somente é possível se houver uma implementação real do sistema em algum formato específico, como é o caso de protótipos, cenários de uso, simulações ou implementação completa do software (MATERA et al., 2002).

De acordo com Valentim (2017), o teste de usabilidade pode ser realizado através de diferentes métodos, a citar:

- Técnicas Baseadas em Observação: têm como premissa observar o usuário interagindo com a interface do software, a fim de compreender o fluxo do usuário e possíveis erros de usabilidade. Uma das técnicas é a *Thinking Aloud*, que tem fundamento verbalizar a interação do usuário com o software, no sentido de que o usuário deve externalizar os seus pensamentos enquanto utiliza o sistema;
- Técnicas Baseadas em Perguntas: consistem em uma série de perguntas, na qual o entrevistador objetiva coletar dados do entrevistado que demonstrem a sua opinião quanto a interface do software. A entrevista pode ser gravada (áudio/vídeo) ou registrada (documento);
- Avaliação Através da Monitoração de Respostas Fisiológicas: é o método mais complexo, que consiste na análise de respostas fisiológicas do usuário no momento de interação com o software. Uma das técnicas adotadas faz uso de aparelhos *Eye Tracking* para determinação do movimento e fixação dos olhos, permitindo, assim, mensurar suas reações no momento de interação e averiguar possíveis erros e demais aspectos interessantes;
- Avaliação Experimental: utiliza estudos controlados e estudos de observações para avaliar aspectos relacionados à interface de um software. À fins didáticos, enfatiza-se que Estudo Controlado refere-se a um estudo direcionado por hipóteses, com posterior análise quantitativa; e que Estudo de Observação se

refere a coleta de dados acerca da tecnologia aplicada, com posterior análise qualitativa.

Cabe salientar que testes de usabilidade são mais custosos do que inspeções de usabilidade, pois demandam tempo dos usuários e, em alguns casos, o uso de espaços adequados para a realização da avaliação, como laboratórios específicos. Esses aspectos negativos são compensados ao se considerar sua maior eficácia na avaliação, uma vez que é considerada a experiência do usuário final (MATERA et al., 2002).

2.6 TRABALHOS RELACIONADOS

Esta seção apresenta subseções com trabalhos, encontrados na literatura, que tem relação com o estudo apresentado neste trabalho.

1) Um Ambiente Virtual com *Feedback* Personalizado para Apoio à Disciplinas de Programação

No trabalho apresentado por Alves e Jaques (2014), os autores buscam uma forma de apoiar as disciplinas de programação, tanto nas questões intrínsecas ao alunos quantos aos professores. Para isto, foi desenvolvido um ambiente virtual denominado Feeper, composto de diferentes recursos, dentre eles: *feedback* personalizado, auxílio ao professor para acompanhamento de seus alunos através de um painel de resultados dos exercícios, dicas sobre erros gerados pelo programa do aluno e adição de marcações no programa para que o aluno aponte para o professor a sua dúvida.

Como forma de validar o ambiente virtual e mensurar o seu potencial de uso, os autores realizaram uma avaliação experimental em uma turma com 24 alunos de uma disciplina de programação do primeiro semestre, do curso de Tecnologia da Informação, em Porto Alegre - RS. Os resultados apontaram grande aceitação das funcionalidades, principalmente para a funcionalidade de envio e recebimento de mensagens ao professor, - funcionalidade esta que obteve a melhor avaliação.

2) Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores

No trabalho apresentado por Paes et al (2013), apontam-se os desafios encontrados no aprendizado de programação - como é o caso da viabilização do processo de resolução de

exercícios de programação e as dificuldades de personalização do ensino - e buscam uma forma automatizada de resolver essas problemáticas.

Tendo em vista esses fatos, os autores realizaram um estudo sobre o software *web The Huxley*, utilizado para apoio à avaliação do aprendizado de alunos em disciplinas de programação de computadores. O SE disponibiliza vários exercícios e permite que os alunos submetam códigos-resposta em diversas linguagens de programação. Após a submissão, um *feedback* com correção automática através de análise sintática do código e testes de aceitação é recebido pelo aluno. Há, ainda, um ranking que lista os alunos com maior quantidade de exercícios resolvidos e um sistema de medalhas. A utilidade do *The Huxley* não restringe-se apenas aos alunos, mas também aos professores, beneficiados com: (i) Visão analítica e fidedigna do desempenho de seus alunos, abrangendo a quantidade de problemas resolvidos, porcentagem de acertos/erros, tipos de problemas com mais erros, detecção de plágio e outros tipos de erros intrínsecos ao aluno; (ii) criação de exercícios; (iii) definição e correção de exercícios e provas; (iv) definição de notas; (v) publicação de resultados.

Por fim, foi realizada uma avaliação para mensurar os impactos do *The Huxley* nas taxas de aprovação ou desistência nas disciplinas iniciais de programação da Universidade Federal de Alagoas - UFAL. Os resultados apontaram um aumento nos índices de aprovação dos alunos após a aplicabilidade do *The Huxley* em contextos estudantis.

3) Uma Ferramenta Gamificada de Apoio à Disciplina Introdutória de Programação

No trabalho apresentado por Campos, Gardiman e Madeira (2015), os autores buscam propor uma ferramenta, denominada de *Kodesh*, capaz de auxiliar o aprendizado de programação de computadores em cursos superiores de Computação.

O software *Kodesh* tem o objetivo de facilitar e estimular a realização de exercícios de programação. Os exercícios disponibilizados pela ferramenta seguem o conteúdo programático das disciplinas iniciais de programação, com sua disseminação ao longo de 18 semanas (período de um semestre acadêmico). Semanalmente, o aluno resolve os exercícios referentes à temática discutida em sala de aula. Em relação aos exercícios, suas resoluções devem ser submetidas para validação. Além do *feedback* proveniente de erros de compilação, o software *web Kodesh* também realiza uma série de testes ocultos com o intuito de generalizar a solução esperada e apresentar ao aluno um *feedback* mais detalhado, com informações dos testes que falharam ou obtiveram sucesso na resolução submetida. Cada exercício resolvido retorna uma pontuação para os alunos, que é utilizada em rankings global e local e aumenta o nível de experiência na ferramenta, desbloqueando diferentes títulos de premiação. Além destas, o software *web*

Kodesh permite que os alunos submetam dúvidas e obtenham respostas de seus colegas - similar a uma rede social. Com a avaliação de aceitação da ferramenta foi constatado um índice de aceitação de 83% pelos usuários.

4) *URI Online Judge Academic*: Integração e Consolidação da Ferramenta no Processo de Ensino/Aprendizagem

Os autores Selivon, Bez e Tonin (2015) buscam apresentar o módulo *Academic*, integrado ao *URI Online Judge* e destinado ao corpo docente. O seu objetivo consiste em auxiliar professores programação, permitindo-lhes acompanhar a evolução e rendimento de seus alunos. Dentre as funcionalidades, destacam-se aqui: (i) controle de listas de exercícios de programação, como determinação dos intervalos de tempo válido para envio da resolução; (ii) acompanhamento dos rendimentos dos alunos, como o histórico de submissões.

Para verificar a aceitabilidade da *URI Online Judge Academic*, os supracitados autores realizaram uma avaliação da aplicabilidade. Os resultados apontaram que os professores utilizaram as ferramentas para realização avaliações presenciais e não-presenciais, acompanhamento mais detalhado dos progressos dos alunos, verificação das dúvidas mais recorrentes e tópicos que necessitam de maior reforço e revisão em sala de aula.

5) *Aprende - Um Novo Sistema Tutor Inteligente para Auxiliar nas Dificuldades dos Conteúdos de Algoritmos e Programação*

No trabalho apresentado por Camargo e Cechinel (2014), os autores buscam propor uma ferramenta capaz de auxiliar o aprendizado de programação de computadores em cursos superiores de Computação. Para isso, foi desenvolvido o APRENDE, um Sistema Tutor Inteligente (STI) para auxiliar alunos com dificuldade a melhorar seu desempenho acadêmico. O sistema possui dois módulos: Aluno e Professor. Para o módulo Aluno, estão dispostas as seguintes funcionalidades: (i) ferramenta para mensurar o estilo de aprendizagem do usuário através do Teste de *Felder*; (ii) exercícios de programação disponibilizados de acordo com seu perfil de aprendizagem; (iii) disponibilidade de materiais de estudos, separados por tópicos e com possibilidade de acesso à usuários não registrados. Para o módulo Professor, estão dispostas as seguintes funcionalidades: (i) submissão de objetos de aprendizagem; (ii) submissão de exercícios de programação. A pesquisa não apresentou dados de avaliações da ferramenta.

6) Mojo: Uma Ferramenta de Auxílio à Elaboração, Submissão e Correção de Atividades em Disciplinas de Programação

No trabalho apresentado por Chaves et al (2013), os autores buscam propor uma ferramenta capaz de auxiliar o aprendizado de programação de computadores em cursos superiores de Computação; porém com foco na elaboração, submissão e correção de atividades práticas de programação. Para isso, desenvolveram uma ferramenta capaz de automatizar o processo de elaboração, submissão e correção de atividades práticas de programação, denominada de Módulo de Integração com os Juízes Online (MOJO).

O MOJO integra Juízes Online ao *Moodle (Modular Object-Oriented Dynamic Learning Environment)* e possui funcionalidades direcionadas ao gerenciamento de recursos e acompanhamento do desempenho dos alunos. Tendo em vista as aplicabilidades das funcionalidades, os desenvolvedores categorizam-nas em dois módulos: Professor e Aluno. Para o módulo do Professor, citam-se as seguintes funcionalidades: (i) exercícios de programação organizados em diferentes classificações; (ii) possibilidade de criar listas de exercícios; (iii) acompanhamento dos desempenhos dos alunos. Para o módulo Aluno, citam-se as seguintes funcionalidades: (i) envio da resolução das questões; (ii) *feedback* do professor.

A pesquisa não apresentou dados de avaliações da ferramenta.

7) Estudo Comparativo

Após buscar os trabalhos relacionados a este estudo, foi realizada uma análise comparativa dos SEs descritos anteriormente, o que fomentou a elaboração de um quadro destacando as principais características dos mesmos, comparando-os à proposta aqui especificada, conforme especificado no Quadro 02.

Quadro 02. Comparação entre trabalhos relacionados e o Loop Academic

Itens de Análise	Nome do SE						
	Feeper	The Huxley	Kodesh	URI Online Judge Academic	APRENDE	MOJO	Loop Academic
Desenvolvido e/ou estudado por	Alves e Jaques (2014)	Paes et al (2013)	Campos, Gardiman e Madeira (2015)	Selivon, Bez e Tonin (2015)	Camargo e Cechinel (2014)	Chaves et al (2013)	Próprio autor
Tipo de SE	Exercícios e Práticas	Exercícios e Práticas	Exercícios e Práticas	Exercícios e Práticas	Exercícios e Práticas	Exercícios e Práticas	Exercícios e Práticas
Categoria de Software	Software Web	Software Web	Software Web	Software Web	Software Web	Plugin	Software Web
Público-Alvo	Alunos e professores de programação	Alunos, professores e monitores da programação	Alunos e professores da programação	Alunos e professores da programação	Alunos e professores da programação	Alunos e professores da programação	Alunos, professores e monitores da programação
Perfis de Usuários	Aluno, Professor	Aluno, Professor, Monitor	Aluno, Professor	Aluno, Professor	Aluno, Professor	Aluno, Professor	Aluno, Professor, Monitor
Linguagem de Programação	Java	Python, C, C++, Java, Octave e Pascal	C	C/C++, Java, Python	Não especificado pelos autores	Não especificado pelos autores	C
Disponibilidade de Exercícios	Contém exercícios	Contém exercícios	Contém exercícios	Contém exercícios	Contém exercícios	Contém exercícios	Contém exercícios
Disponibilidade de Feedback	Feedback personalizado com base nos cenários de teste cadastrados pelo professor	Envio de dúvidas à professores e monitores, com resposta personalizada	Feedback de erros de compilação	Feedback de erros de compilação	O trabalho não menciona	Relacionado ao Juiz Online (JO) integrado	Envio de dúvidas à professores e monitores, com resposta personalizada
				Indicação do percentual de casos de teste que falharam			Feedback de erros de compilação
Disponibilidade de Recursos de Ajuda	Emite dicas sobre os erros gerados pelo programa do aluno, permitindo que este consiga evoluir na correção do seu programa	O trabalho não menciona	Consulta de dicas pré-cadastradas por professores	O trabalho não menciona	Materiais de estudo, inclusive para usuários não cadastrados	O trabalho não menciona	Consulta de dicas e códigos de apoio pré-cadastrados por professores e monitores, materiais de apoio, fóruns de discussão
Avaliação de Usabilidade	O trabalho não menciona	O trabalho não menciona	O trabalho não menciona	O trabalho não menciona	O trabalho não menciona	O trabalho não menciona	Níveis de Acerto, Tempo de Realização de Atividades e Qualidade Afetiva
Avaliação Geral da Ferramenta	Avaliação Quantitativa de Experiência de Uso	Estudo de caso em turmas de programação da Universidade Federal de Alagoas	Avaliação de Aceitação	Avaliação de aspectos positivos da interface	Estudo de caso	O trabalho não menciona	A ser realizada
Acesso	www.feeper.unisinos.br	www.thehuxley.com	www.kodesh.ufrr.br	www.urionlinejudge.com.br/academic	O trabalho não menciona	Plugin a ser instalado no Moodle www.greenbeans.com.br/osvaldomesquita/mojo/	Protótipo acessível através de:
			Indisponível até o dia 24/02/2019				www.sites.google.com/view/repositorioloopacademic

Fonte: Autoria própria

Diante dos dados apresentados no Quadro 02, enfatiza-se que o *Loop Academic* diferencia-se dos demais por: (i) prover acesso a dicas e código de apoio; (ii) fornecer materiais de apoio destinados aos estudos das temáticas da programação, com maior direcionamento àquelas que o aluno tem dificuldade, e estando esses caracterizados como artefatos produzidos pelo LABIE (Laboratório de Pesquisa em Informática na Educação) e Programming Loop; (iii) permitir que o aluno participe de fóruns de discussão, destinados a promover a disseminação de conhecimentos e retirada de dúvidas; (iv) aplicar avaliação de usabilidade, estando essa atividade dividida em três etapas complementares: níveis de acerto, tempo de realização de atividades e qualidade afetiva.

3 METODOLOGIA

Esta seção descreve as fases inerentes à metodologia desenvolvida neste trabalho. Na seção 3.1 descrevemos a realização de uma Pesquisa de Opinião: um *Survey*. A seção 3.2 apresenta o planejamento para a prototipação, elicitación de requisitos, definição dos *personas*, protótipos de baixa fidelidade e ferramenta utilizada para elaboração do protótipo de alta fidelidade. A seção 3.3 apresenta o planejamento para o teste de usabilidade.

3.1 PESQUISA DE OPINIÃO: UM *SURVEY*

Para compreender e validar a viabilidade de uma ferramenta de apoio ao ensino e aprendizagem de programação no ambiente do curso de Bacharelado em Tecnologia da Informação (BTI), *Campus Pau dos Ferros - UFERSA*, foi realizado um *survey* direcionado a alunos, professores e monitores da disciplina de Algoritmos (disciplina que aborda os conteúdos de programação introdutória do referido curso superior). O questionário eletrônico foi aplicado via *Internet* e divulgado via correio eletrônico e redes sociais. As perguntas e as opções de resposta foram cadastradas a partir dos recursos do *Google Forms* (formulário do Google).

O *survey* direcionado aos alunos (ver Quadro 03) foi organizado em quatro seções, estando cada qual organizada da seguinte forma:

- Seção inicial: apresentação do propósito da pesquisa e consentimento dos alunos quanto à utilização de suas respostas no desenvolvimento do protótipo;
- Dados do Aluno: seção que consistiu na coleta dos dados do aluno no que diz respeito às disciplinas de programação introdutória. Enfatiza-se que todas as respostas foram de caráter anônimo;
- Plataforma *Web*: seção que consistiu na coleta dos dados referentes à plataforma *web*, como avaliação de sua utilidade e possíveis requisitos funcionais;
- Sugestões: seção que consistiu na coleta de sugestões que não foram enfatizadas no questionário eletrônico, permitindo ao aluno direcionar uma sugestão de melhoria.

O *survey* ficou disponível do dia 20 de dezembro de 2018, às 20h36min, até 17 de janeiro de 2019, às 15h14min, e contou com a participação de 28 alunos. Os resultados obtidos são apresentados e discutidos na Subseção 4.1.

Quadro 03. Questionamentos do *survey* aplicado com alunos

Nº	Questionamento	Tipo	Alternativas ou Respostas
01	Das disciplinas abaixo, em qual delas você está matriculado atualmente?	Múltipla escolha	I. Algoritmos/Laboratório de Algoritmos; II. Pré-Algoritmos.
02	Você está cursando a disciplina citada anteriormente	Múltipla escolha	I. Pela 1ª vez; II. Pela 2ª vez; III. Pela 3ª vez; IV. Pela 4ª vez; V. Pela 5ª vez; VI. Não quero informar.
03	Você considera que uma plataforma <i>web</i> com conteúdos relacionados à disciplina de Algoritmos/Pré-Algoritmos seria útil para te ajudar a estudar programação?	Escala Linear	I. 1 para Considero Totalmente; II. 2 para Considero Razoavelmente; III. 3 para Indiferente; IV. 4 para Desconsidero Razoavelmente; V. 5 para Desconsidero Totalmente.
04	Você considera que uma plataforma <i>web</i> de resolução de exercícios de Algoritmos seria útil para te ajudar a estudar?		
05	Como a plataforma <i>web</i> poderia lhe ser útil?	Caixas de Seleção	I. Fornecendo dicas para resolver exercícios (questões); II. Informando o nível de dificuldade de cada exercício para que eu possa começar pelo exercício que está relacionado ao meu nível de conhecimento; III. Apresentando exemplos de códigos de programação relacionados ao conteúdo que estou estudando; IV. Dispondo materiais de apoio ao estudo de programação (como apostilas, livros etc.); V. Dispondo recursos de apoio ao estudo de programação (como vídeo-aulas etc.); VI. Permitindo que eu envie dúvidas ao professor ou monitor pela própria plataforma; VII. Permitido minha participação em fóruns de discussão; VIII. Não sei responder.
06	Há alguma outra sugestão que você gostaria de expor, mas que não foi contemplada neste formulário de pesquisa?	Parágrafo	Resposta livre.

Fonte: Autoria própria

Enfatiza-se que os resultados obtidos com os *surveys* direcionados aos professores e monitores estão disponíveis no repositório *online* de arquivos do *Loop Academic*², pois não chegaram a ser utilizados como base para o desenvolvimento desta pesquisa por limitações de tempo. Estes servirão, portanto, para a elaboração de trabalhos futuros.

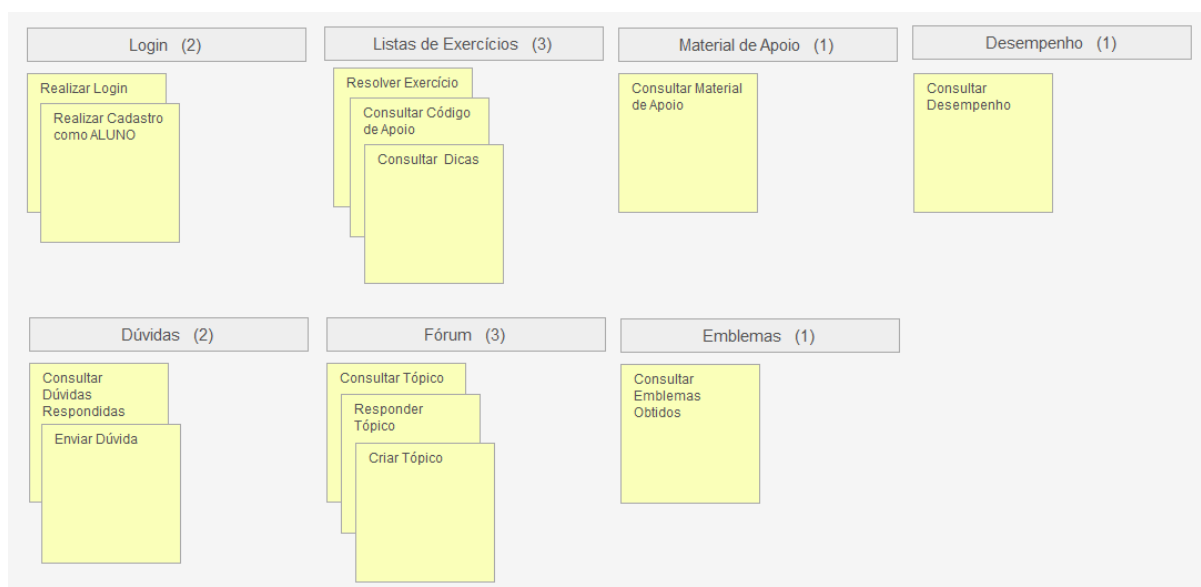
3.2 PLANEJAMENTO PARA A PROTOTIPAÇÃO

3.2.1 Elicitação e Especificação dos Requisitos

O processo de elicitação e especificação dos requisitos fez uso da técnica de *Card Sorting*. Com o software *Sticky Sorting*³, foi possível agrupar as principais funcionalidades obtidas com o *survey* (Subseção 3.1). A Figura 01 apresenta o *Card Sorting* realizado.

Note que as funcionalidades definidas se encontram organizadas em 7 (sete) grupos: (i) login; (ii) listas de exercícios; (iii) material de apoio; (iv) desempenho; (v) dúvidas; (vi) fórum; (vii) emblemas⁴. Cada grupo e suas respectivas funcionalidades foram tomados como base para a elaboração dos documentos que caracterizam os requisitos do sistema: (i) requisitos funcionais e não funcionais; (ii) casos de uso.

Figura 01. *Card Sorting*



Fonte: Autoria própria, 2019

² Acessível através de: www.sites.google.com/view/repositorioloopacademic>.

³ Demais informações e download: www.chip.de/downloads/StickySorter_49391737.html

⁴ Figura simbólica, geralmente utilizada em jogos para indicar que o jogador obteve alguma conquista.

3.2.2 Definição dos Personas

O processo de criação de Personas foi realizado pelo *Make My Persona Tool*⁵, de autoria da *HubSpot*. A ferramenta é gratuita e consiste em um gerador de personas com diversas características relevantes à um persona. A escolha do *Make My Persona Tool* se deu por sua facilidade na concepção de um persona.

Para guiar o processo de elaboração dos protótipos, foram criados dois personas para representar os perfis de futuros usuários da plataforma *web*: aluno novato e aluno veterano e repetente. Para isso, foram considerados os seguintes aspectos: idade, escolaridade, uso/acesso à redes sociais, ocupação, objetivos, problemas e como a plataforma pode ajudá-lo. Assim, foram criados os personas: Tiago Include, o novato desamparado; e, Tereza Return, a veterana focada. Os personas e suas características são apresentados e discutidos na seção de resultados (Seção 4.3).

3.2.3 Prototipação de Baixa Fidelidade e de Alta Fidelidade

A partir dos requisitos e dos personas apresentados e discutidos, respectivamente, nas Subseções 3.2.1 e 3.2.2, a prototipação do SE teve início. Tendo em vista o escopo inicial, realizou-se a prototipação de baixa fidelidade com o intuito de explorar aspectos da interface em conjunto com um esboço inicial das funcionalidades.

O processo de elaboração dos protótipos de baixa fidelidade foi realizado de forma a enfatizar as telas principais, pois, a partir destas se tornaria possível construir as demais telas características do SE discutido neste trabalho.

Com base nos protótipos de baixa fidelidade, foram elaborados os protótipos de alta fidelidade, tendo em vista a sua representação do SE a ser construído e a sua necessidade para a aplicação do teste de usabilidade.

3.2.4 Definição das Ferramentas

Para a construção dos protótipos de alta fidelidade, realizou-se uma pesquisa prévia acerca de possíveis ferramentas utilizadas para prototipação. Como resultado, foram encontradas as seguintes ferramentas: *InVision*, *HotGloo*, *SketchApp* e *Adobe XD*.

⁵ Demais informações e uso: www.hubspot.com/make-my-persona

O software de prototipação *InVision*⁶ é grátis e possui funcionalidades intuitivas para criação de ideias, *design*, protótipos e gerenciamento de *design*. A concepção das telas está baseada em clicar e arrastar para adicionar diferentes pontos de acesso e assim transformar projetos estáticos em projetos interativos. O *InVision* permite criar itens de aplicabilidade global, como menus, e adicionar gestos e animações para transições. Outros aspectos positivos da ferramenta estão em sua facilidade de compartilhar os protótipos com os *stakeholders* através do *Browser* ou através de *download* para execução em dispositivos mobile; em sua compatibilidade com qualquer ferramenta de *design*, o que permite upload de arquivos JPG, PNG, PDF ou GIF; e em sua integração com o *DropBox*, garantindo sincronização e garantia de segurança do protótipo.

O software de prototipação *HotGloo*⁷ é uma ferramenta de experiência de usabilidade, *wireframe* e prototipagem destinada à construção de protótipos de softwares *web*, *mobile* e *wearebles*. Suas funcionalidades incluem mecanismos adequados para criar e compartilhar protótipos interativos com *stakeholders*. De acordo com os desenvolvedores, o *HotGloo* foi projetado com uma baixa curva de aprendizado, o que leva o usuário a despendar menor tempo em aprender a execução de suas funcionalidades; e é otimizado para dispositivos *mobile*, permitindo edição, revisão e teste dos protótipos. O *HotGloo* é um software pago, com 7 dias grátis de testabilidade.

O software de prototipação *SketchApp*⁸ é uma ferramenta de prototipagem destinada à construção de protótipos de softwares *web* e *mobile*. Suas funcionalidades são similares às dos softwares de prototipação citados anteriormente. Seus diferenciais incluem precisão na edição de *pixels*, edição de vetores, exportação de códigos e grades. Por ser todo em vetor, adaptar projetos para diferentes resoluções de tela é mais prático. Outro aspecto positivo é a sua limitação com código CSS, o que garante que o *design* gráfico seja realizado por um desenvolvedor *front-end* sem muitas dificuldades. O *SketchApp* é um software pago, com 30 dias de testabilidade.

O software de prototipagem *Adobe XD* é uma ferramenta flexível e rápida para criação de protótipos que envolvem o *design* de interface e a experiência de usuário. Suas funcionalidades permitem, além da criação dos protótipos, animar transições de telas; visualizar e testar protótipos em qualquer dispositivo - *Web*, *Android* e *IOS*; transições temporizadas; e

⁶ Demais informações e download: www.invisionapp.com

⁷ Demais informações e download: www.hotgloo.com

⁸ Demais informações e download: www.sketchapp.com

publicar e avaliar o protótipo perante os *stakeholders*, o que facilita a coleta de *feedback*. O *Adobe XD* possui um plano gratuito, que permite ao profissional de *design* compartilhar um protótipo ativo e armazenar até 2 GB em nuvem. O seu plano pago possui maior abrangência, porém é inadequado à este trabalho tendo em vista o escopo aqui abordado.

Para a elaboração dos protótipos de alta fidelidade deste trabalho, foi utilizado o software de prototipagem *Adobe XD*⁹. Ressalta-se que a sua escolha se deu por conta da pequena curva de aprendizado estabelecida pela ferramenta de prototipagem, o que é possível graças à sua interface minimalista e direcionada ao seu propósito de uso. Além dessa, a capacidade de realizar a avaliação da interface através de um ambiente *web*, com capacidade de coleta de opiniões e marcações de irregularidades ou pontos em destaque na própria interface, foi um aspecto que teve grande peso na escolha da ferramenta, tendo em vista a avaliação de usabilidade a ser realizada. Também cita-se o armazenamento em nuvem, que conferiu maior segurança ao protótipo desenvolvido.

3.3 PLANEJAMENTO E EXECUÇÃO DO TESTE DE USABILIDADE

Com a finalidade de avaliar a usabilidade do protótipo de alta fidelidade, o teste de usabilidade realizado abrangeu três etapas: (i) nível de acertos ao realizar determinadas tarefas; (ii) tempo de realização de cada tarefa; (iii) qualidade afetiva da interface, com aplicação da Escala SAM (*Self Assessment Manikin*), proposta por Lang (1980), cuja intuito é mensurar a qualidade afetiva de uma interface.

Para a execução da primeira etapa do teste de usabilidade foram selecionadas 7 (sete) tarefas estratégicas, que permitiram a exploração das funcionalidades do SE prototipado. Neste sentido, foram definidas as seguintes tarefas:

- Tarefa 01: realizar login;
- Tarefa 02: responder exercício 07 da Lista de Exercícios 02 - Estruturas de Decisão;
- Tarefa 03: consultar material de apoio sobre Estruturas de Decisão;
- Tarefa 04: consultar desempenho na resolução da Lista de Exercícios 02 - Estruturas de Decisão;
- Tarefa 05: enviar dúvida;
- Tarefa 06: responder tópico no fórum;

⁹ Demais informações e download: www.adobe.com/br/products/xd.html

- Tarefa 07: consultar emblemas obtidos.

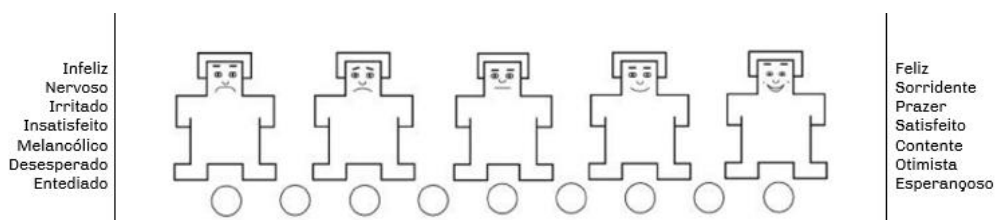
Para cada tarefa elencada anteriormente, foi definido um cenário fictício com o intuito de orientar a realização da tarefa. Cada cenário teve o menor grau possível de detalhes, pois o intuito era levar o usuário à exploração do protótipo para realizar a tarefa proposta. Após a concretização de cada tarefa, foi questionado o seu nível de acertos.

De acordo com Valentim (2018), o nível de acertos é uma avaliação que consiste em mensurar o quantitativo de usuários que obtiveram ou não obtiveram êxito na realização de uma tarefa. Para tal, são adotados os critérios: (i) *Sucesso-Fácil*, no qual o usuário afirma ter concluído a tarefa em sua primeira tentativa; (ii) *Sucesso-Difícil*, no qual o usuário afirma ter concluído a tarefa com bastante dificuldade; (iii) *Insucesso*, no qual o usuário afirma não ter sido capaz de concluir a tarefa ou ter desistido de concluí-la.

Para a execução da segunda etapa do teste de usabilidade, o tempo de realização de cada tarefa pelo usuário foi cronometrado e, em seguida, registrado.

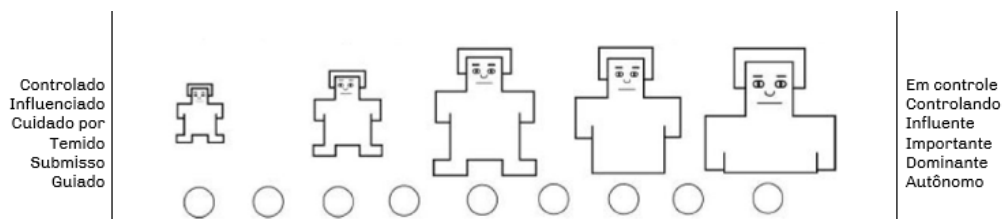
Para a execução da terceira etapa do teste de usabilidade, foi utilizada a escala SAM, que leva em consideração três dimensões em sua avaliação: (i) Satisfação, que consiste em medir o nível de prazer/desprazer; (ii) Controle, que consiste em medir o nível de controle - o usuário informa se se sentiu controlado ou se obteve controle; (iii) Motivação, que consiste em medir o quanto o usuário se sentiu motivado. As Figuras 02, 03 e 04 apresentam, respectivamente, as escalas SAM adotadas para as dimensões Satisfação, Controle e Motivação.

Figura 02. Escala SAM: Satisfação



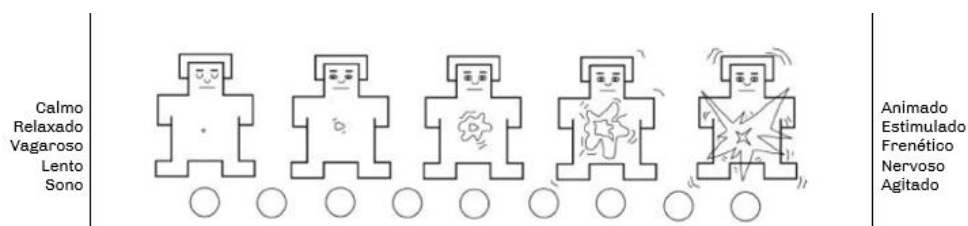
Fonte: Adaptado de Lang (1980)

Figura 03. Escala SAM: Controle



Fonte: Adaptado de Lang (1980)

Figura 04. Escala SAM: Motivação



Fonte: Adaptado de Lang (1980)

O teste de usabilidade foi aplicado através de um questionário *online*, denominado Avaliação de Usabilidade - Erros por Atividade e Escala SAM, com acompanhamento dos entrevistados. O questionário citado pode ser consultado no repositório de arquivos do *Loop Academic*¹⁰.

A participação no teste de usabilidade foi voluntária. Todos os participantes assinaram um Termo de Consentimento Livre e Esclarecido (TCLE), afirmando que concordavam em participar das avaliações. O termo de consentimento utilizado pode ser encontrado no Anexo A.

O Teste de Usabilidade foi realizado em um ambiente de baixo custo, tendo em vista a indispensabilidade de equipamentos extras, tais como laboratórios de usabilidade. Foi necessário, portanto, um computador com o protótipo de alta fidelidade e com acesso à internet, um cronômetro e uma planilha eletrônica para que fossem anotados os tempos de realização de cada atividade.

Durante a realização do teste, foram demonstradas aos participantes as atividades a serem realizadas e como responder o questionário. No tocante à dúvidas, somente foram sanadas aquelas capazes de não comprometer a veracidade do teste realizado. Comentários e erros dos participantes do teste de usabilidade foram anotados.

¹⁰Acessível através do link: www.sites.google.com/view/repositorioloopacademic.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Nesta seção são analisados e discutidos os resultados obtidos com esta pesquisa. A disposição das informações perfaz uma linha de raciocínio gradativa, i.e., na Subseção 4.1 são apresentados e discutidos os resultados do *survey*, que serviu como ponto inicial para a elaboração deste trabalho. Em seguida, são apresentados, na Subseção 4.2, os requisitos funcionais que caracterizam o *Loop Academic*. A Seção 4.3 apresenta as principais telas dos protótipos de baixa e alta fidelidades. E, por fim, apresenta-se, na Subseção 4.4, os resultados do teste de usabilidade para as etapas nível de acertos, tempo de execução e qualidade afetiva.

4.1 RESULTADOS DO SURVEY

Nesta seção são discutidos e apresentados os resultados mais relevantes do formulário aplicado com 28 (vinte e oito) alunos - novatos e não-aprovados - das disciplinas de Algoritmos e Laboratório de Algoritmos, do curso de BTI da UFERSA, *Campus Pau dos Ferros*. A totalidade dos resultados são apresentados no repositório de arquivos do *Loop Academic*¹¹.

Quando indagados acerca do número de vezes em que cursaram as disciplinas de Algoritmos e/ou Laboratório de Algoritmos, 17 alunos (60,7%) afirmaram estarem cursando as disciplinas iniciais de programação pela primeira vez; 8 alunos (28,6%) afirmaram estarem cursando pela 2ª vez; 2 alunos (7,1%) afirmaram estarem cursando pela 3ª vez; e 1 aluno (3,6%) afirmou estar cursando pela 4ª vez. Apesar dos dados não refletirem por completo as turmas das disciplinas iniciais de programação, tem-se que o quantitativo de alunos repetentes (39,3%, 11 alunos) é próximo ao quantitativo de alunos novatos (60,7%, 17 alunos). O Gráfico 01 apresenta os resultados citados, que caracterizam o perfil dos alunos participantes do *survey* inicial.

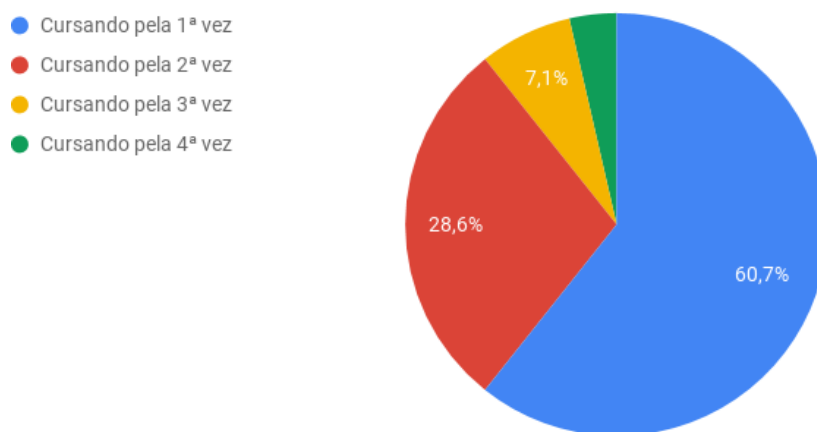
Para responder os próximos dois questionamentos, os alunos tiveram à sua disposição uma escala com valores de 1 a 5, que refletia as seguintes avaliações: (1) considero totalmente; (2) considero razoavelmente; (3) não considero, não desconsidero; (4) desconsidero razoavelmente; (5) desconsidero totalmente.

Ao serem questionados sobre a utilidade de uma plataforma *web* com conteúdos relacionados à disciplina de Algoritmos/Pré-Algoritmos para o estudo de programação (Gráfico 02), 15 alunos (53,6%) a consideraram totalmente útil, 7 alunos (25%) a consideraram razoavelmente útil, 4 alunos (14,3%) ficaram indecisos quanto à sua utilidade e 2 alunos (7,1%)

¹¹ Acessível através do link: www.sites.google.com/view/repositorioloopacademic

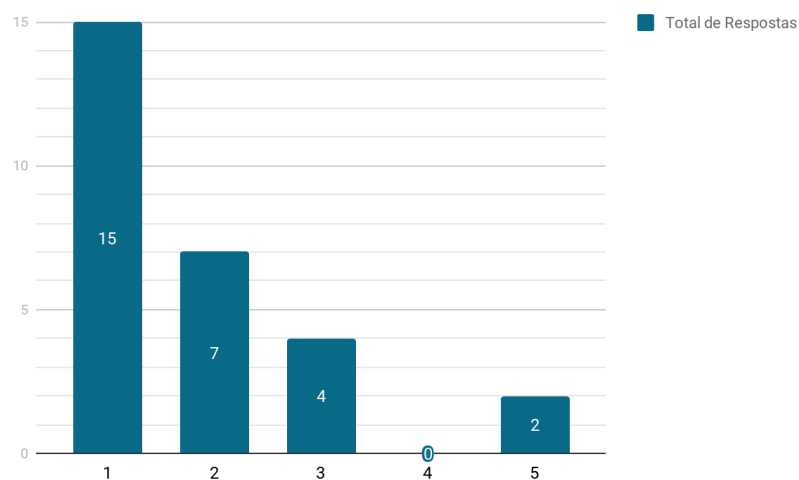
não viram utilidade na plataforma, conforme pode ser visto no Gráfico 02. Diante dos dados, 22 alunos (78,6%) viram algum tipo de utilidade na plataforma sugerida.

Gráfico 01. Tipos de aluno de Algoritmos e Laboratório de Algoritmos no período 2018.2



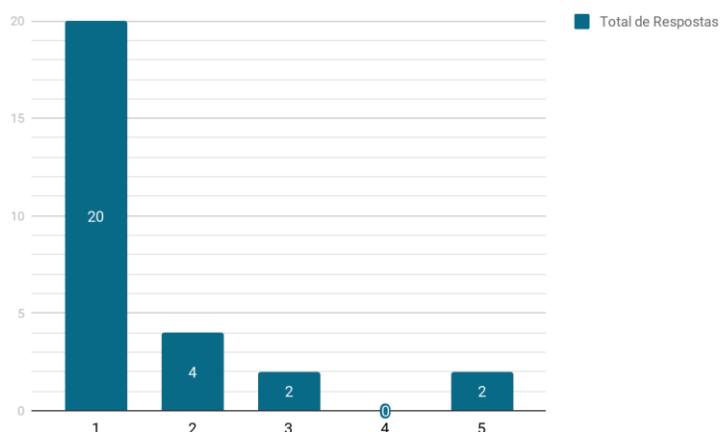
Fonte: Autoria própria.

Gráfico 02. Utilidade do *Loop Academic* para o estudo de Algoritmos



Fonte: Autoria própria.

Ao serem questionados sobre a utilidade de uma plataforma *web* de resolução de exercícios de Algoritmos (Gráfico 03), 20 alunos (71,4%) a consideraram totalmente útil, 4 alunos (14,3%) a consideraram razoavelmente útil, 2 alunos (7,1%) ficaram indecisos quanto à sua utilidade e 2 alunos (7,1%) não viram utilidade na plataforma, conforme pode ser visto no Gráfico 03. Diante dos dados, 24 alunos (85,7%) viram utilidade na plataforma sugerida, o que também é uma justificativa relevante para a concepção deste trabalho e demonstra que uma grande parcela dos alunos muito provavelmente utilizará o SE proposto neste trabalho.

Gráfico 03. Utilidade do *Loop Academic* para o incentivo aos estudos

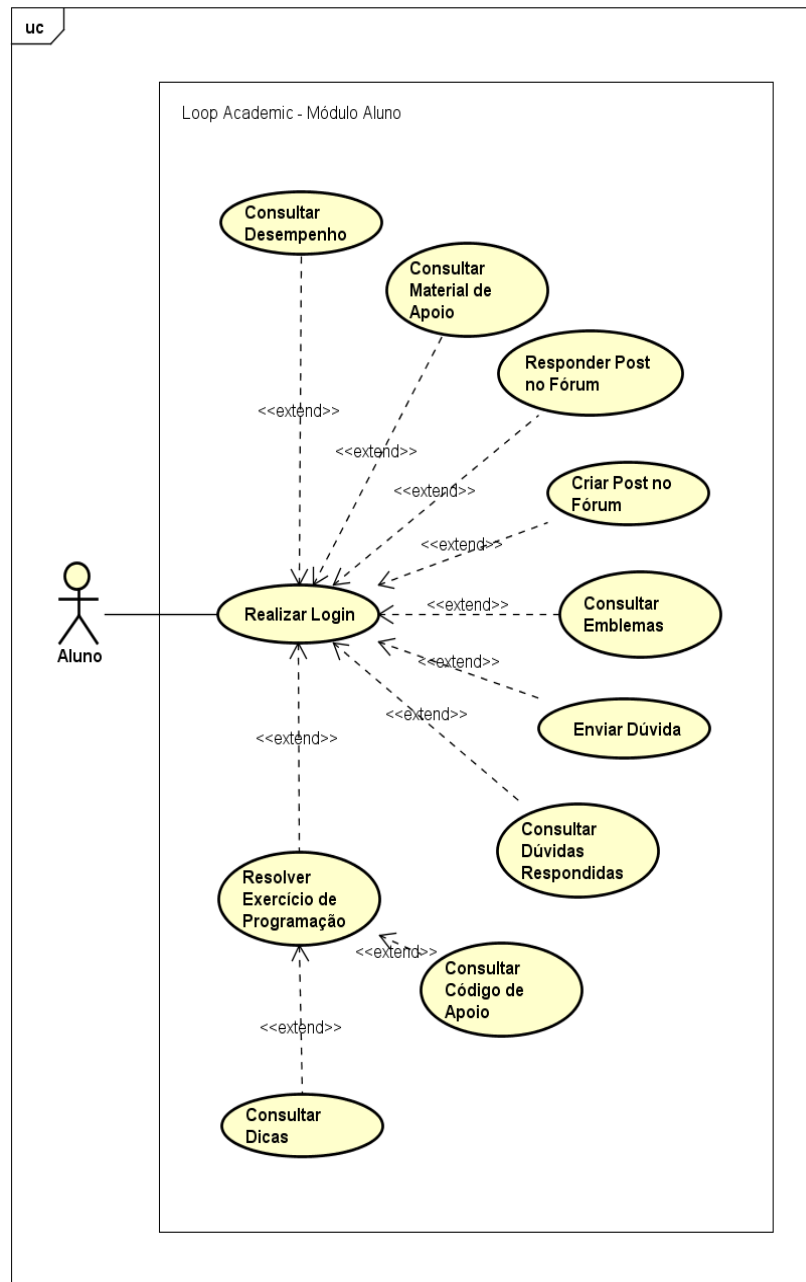
Fonte: Autoria própria.

Os alunos que responderam o formulário tiveram ao seu dispor uma série de funcionalidades a serem escolhidas para o SE, de acordo com a sua própria concepção de utilidade. Suas escolhas refletiram que: (i) 17 alunos (60,7%) consideram útil que a plataforma forneça dicas para a resolução dos exercícios; (ii) 11 alunos (39,3%) consideram útil que a plataforma informe o nível de dificuldade do exercício para que o aluno resolva questões equivalentes ao seu grau de conhecimento; (iii) 19 alunos (67,9%) consideram útil que a plataforma apresente exemplos de código de programação relacionados ao conteúdo que está sendo estudado; (iv) 10 alunos (35,7%) consideram útil que a plataforma *web* disponha de materiais de apoio para o estudo de programação, como apostilas, livros etc.; (v) 12 alunos (42,9%) consideram útil que a plataforma *web* disponibilize recursos multimídia para o estudo de programação, como vídeo-aulas etc.; (vi) 14 alunos (50%) consideram útil que a plataforma *web* permita que dúvidas sejam enviadas ao professor e ao monitor.

4.2 REQUISITOS FUNCIONAIS

Os dados obtidos na Subseção 4.1 foram relevantes para a elaboração dos requisitos funcionais do *Loop Academic*. A Figura 05 apresenta o diagrama de casos de uso que demonstra as funcionalidades primordiais para o funcionamento do SE. Enfatiza-se a presença dos documentos de requisitos e de casos de uso no repositório de arquivos do *Loop Academic*¹², caso o leitor tenha necessidade de compreender melhor a organização do SE citado.

¹² Acessível através do link: www.sites.google.com/view/repositorioloopacademic

Figura 05. Diagrama de Casos de Uso do *Loop Academic* - módulo aluno

Fonte: Autoria própria, 2019

4.3 PERSONAS

A seguir, são descritos e apresentados os dois personas definidos para o escopo deste trabalho. Enfatiza-se que a concepção de seus perfis, objetivos, problemas e auxílio do *Loop Academic* na resolução de seus problemas e alcance de seus objetivos guiou a etapa de desenvolvimento dos protótipos de baixa e alta fidelidade, estando os protótipos de alta fidelidade descritos e apresentados na Seção 4.4 deste trabalho.

A Figura 06 apresenta o primeiro persona criado, denominado de “Tiago Include, o novato desamparado”.

Figura 06. Persona 01: Tiago Include, o novato desamparado

Tiago Include, o novato desamparado



Age
18 anos

Highest Level of Education
Superior incompleto

Social Networks








Ocupação
Estudante do curso de Bacharelado em Tecnologia da Informação (BTI) na Universidade Federal Rural do Semi-Árido (UFERSA)

Objetivos
Aprender e aprimorar suas habilidades na programação
Passar na disciplina Algoritmos

Problemas
Como é novato no curso, Tiago Include enfrenta os seguintes problemas:

1. Dificuldades para compreender os conteúdos da disciplina Algoritmos;
2. Dificuldade para encontrar materiais adequados aos seus níveis de conhecimento;
3. Dificuldade para resolver as listas de exercícios da disciplina;
4. Vergonha em tirar dúvidas com o professor;
5. Indisponibilidade de participar das atividades da monitoria, devido ao fato de residir em outra cidade.

Como o Loop Academic pode ajudá-lo?
Fazendo uso dos recursos disponibilizados pelo Loop Academic, o Tiago Include pode:

1. Resolver os exercícios das listas disponibilizadas pelo seu professor em um ambiente direcionado ao foco, com possibilidade de consulta de dicas e código de apoio;
2. Ter acesso à materiais de apoio direcionados a auxiliá-lo no estudo das temáticas de Algoritmos das quais ele tem maior dificuldade ou queira aprimorar seu conhecimento;
3. Enviar dúvidas de forma anônima ao professor e monitor da disciplina.

Fonte: Autoria própria

Os problemas enfrentados pelo Tiago Include compreendem as problemáticas encontradas no processo de ensino-aprendizagem de programação inicial em cursos de nível superior, dos quais foram citados nos trabalhos dos autores Pereira Júnior et al. (2005), Raposo e Dantas (2016), e Holanda, Coutinho e Fontes (2018) na Subseção 2.1. deste trabalho. Ressalta-se que os problemas enfrentados pelo respectivo persona também foram fundamentados de acordo com a experiência dos autores no projeto de Monitoria de Algoritmos da universidade, objeto de estudo que permitiu um maior conhecimento das reais problemáticas enfrentadas pelos alunos iniciantes na programação.

A Figura 07 apresenta o segundo persona criado, denominada de “Tereza Return, a veterana focada”. Os problemas enfrentados pela Tereza Return diferem em alguns aspectos dos problemas enfrentados pelo Tiago Include.

Figura 07. Persona 02: Tereza Return, a veterana focada



Fonte: Autoria própria

4.4 PROTÓTIPOS DESENVOLVIDOS

A discussão aqui realizada tem como premissa apresentar e discutir as principais telas dos protótipos de baixa e alta fidelidades. Antes de sua apresentação, são demonstrados alguns componentes de relevância ao desenvolvimento dos protótipos.

4.4.1 Componentes Iniciais

O software educacional recebeu o nome de *Loop Academic*, em referência ao canal do *YouTube Programming Loop*, no qual estão armazenadas as videoaulas produzidas pelos autores deste trabalho - e que também serão utilizadas como componentes dos materiais de apoio. Partindo dessa premissa, foi produzida uma logomarca para o *Loop Academic* (Figura 08) e definidos um endereço fictício (“www.loopacademic.com.br”) e um *favicon* (Figura 09) - pequenos ícones associados à uma página *web*.

Figura 08. Logomarca do *Loop Academic*



Fonte: Autoria própria

Figura 09. Favicon do *Loop Academic*



Fonte: Autoria própria

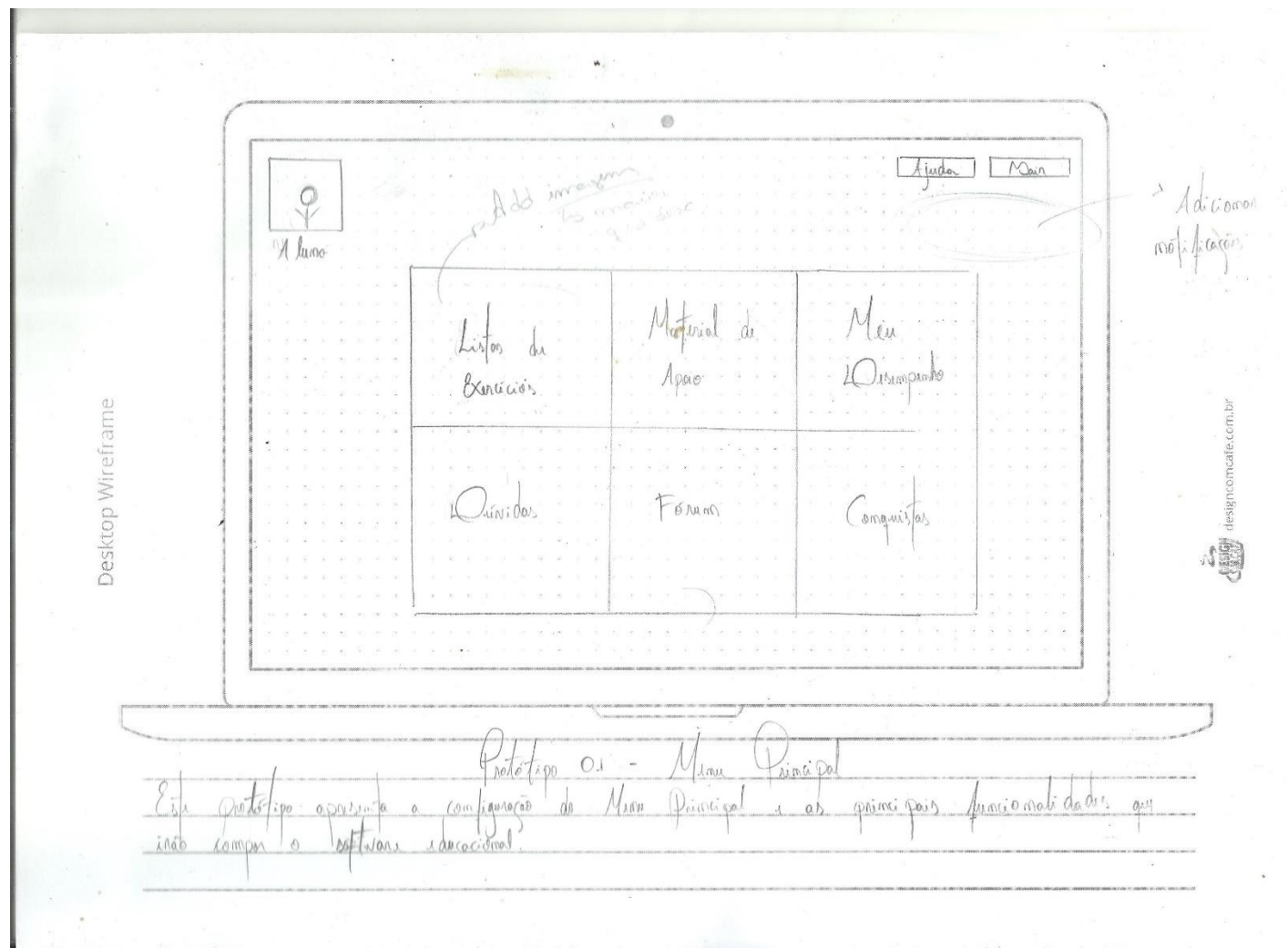
4.4.2 Telas do Protótipo de Baixa Fidelidade

A concepção das telas do protótipo de baixa fidelidade do *Loop Academic* seguiram: (i) as heurísticas de usabilidade definidas na Subseção 2.5 (Avaliação de Interfaces); (ii) os requisitos funcionais obtidos do *survey* realizado com os alunos iniciantes do curso de Bacharelado em Tecnologia da Informação na Universidade Federal Rural do Semi-Árido (UFERSA), *Campus* Pau dos Ferros, presentes na Subseção 4.2 (Requisitos Funcionais).

Apresenta-se, a seguir, as telas que compõem o protótipo de baixa fidelidade. Enfatiza-se que por redução do escopo, somente serão apresentadas as principais telas. As demais podem ser consultadas e melhor analisadas no repositório de arquivos do *Loop Academic*¹³.

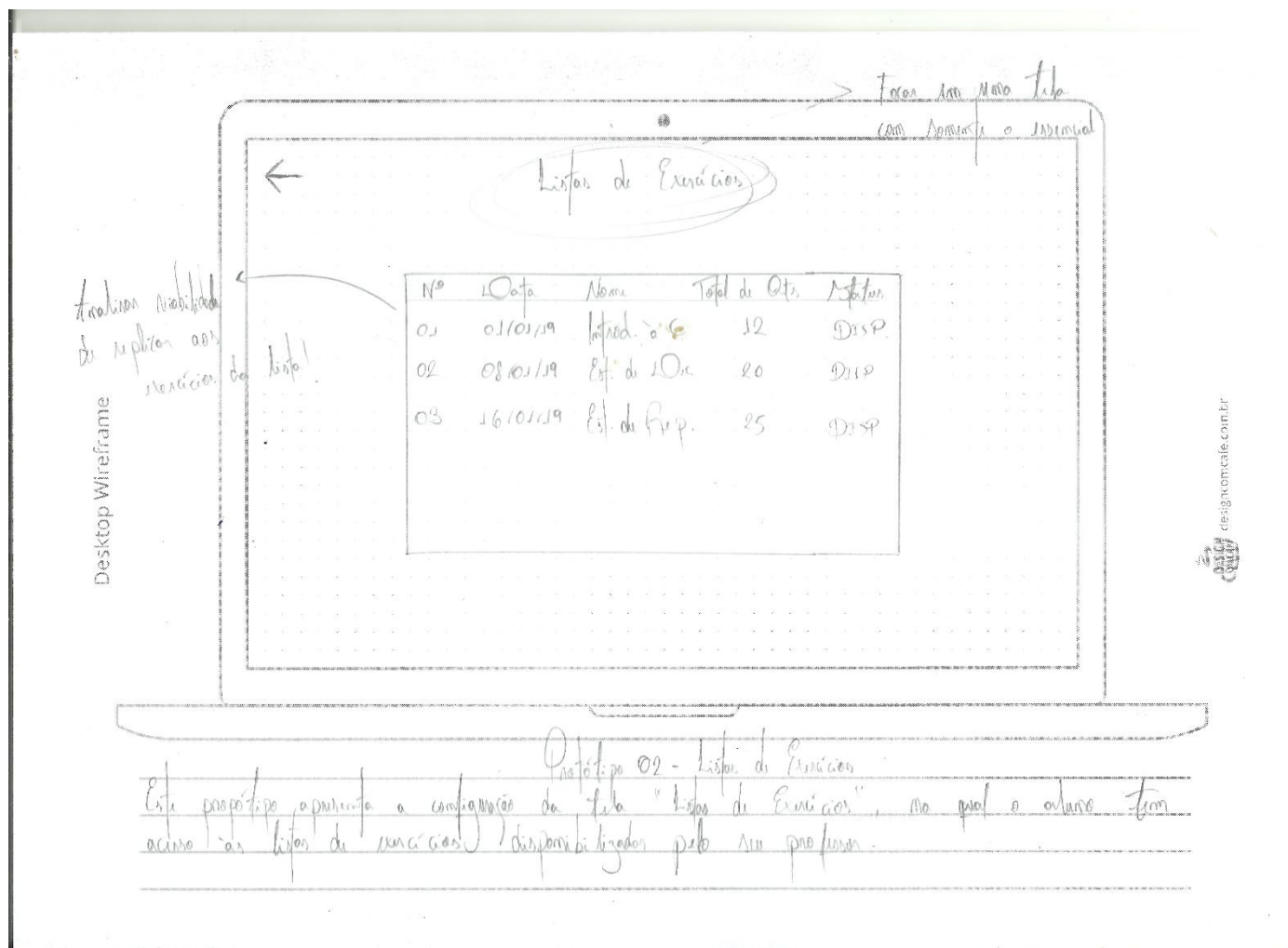
¹³Acessível através do link: www.sites.google.com/view/repositorioloopacademic

Figura 10. Protótipo de baixa fidelidade: menu principal



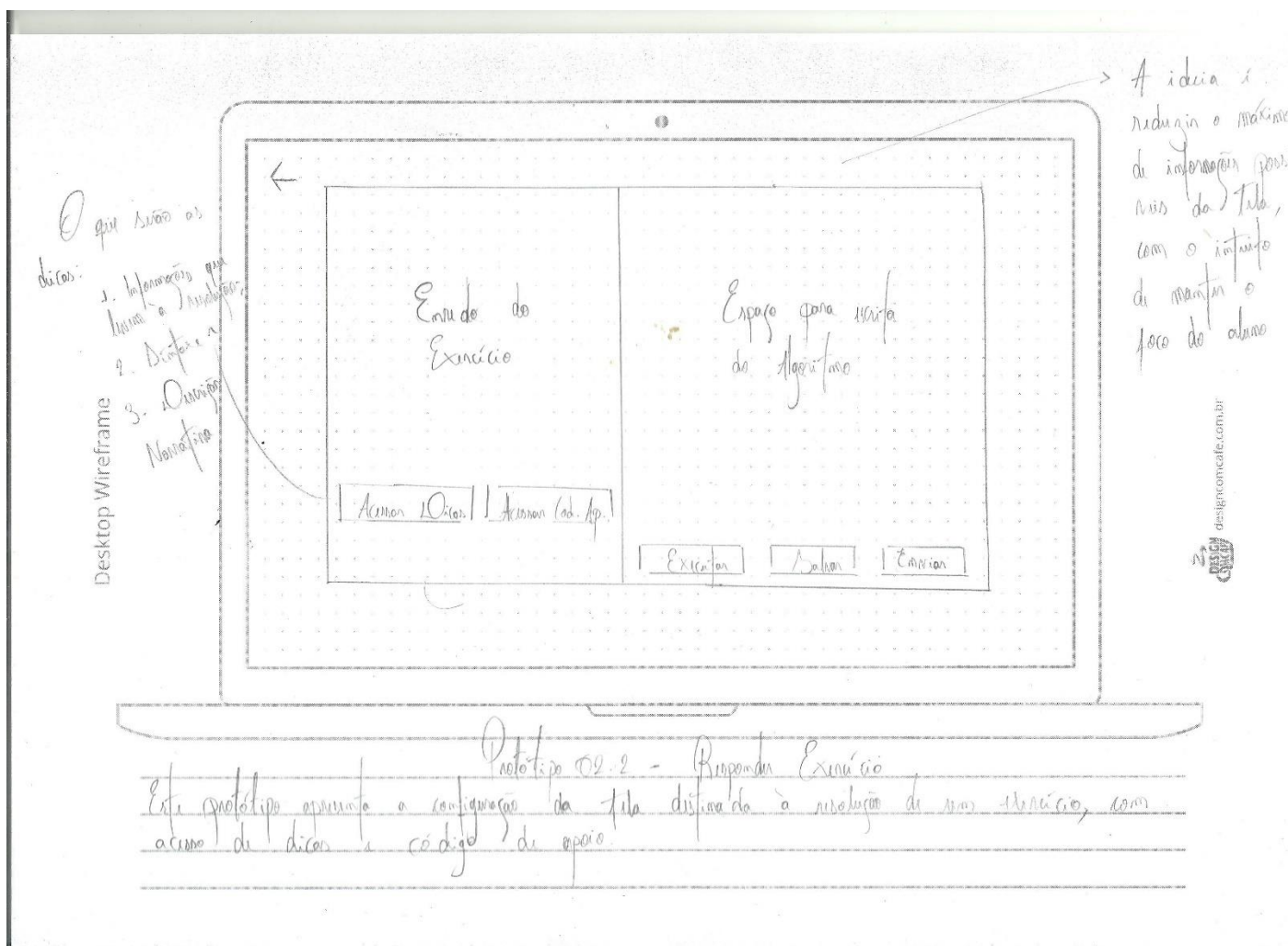
Fonte: Autoria própria

Figura 11. Protótipo de baixa fidelidade: listas de exercícios



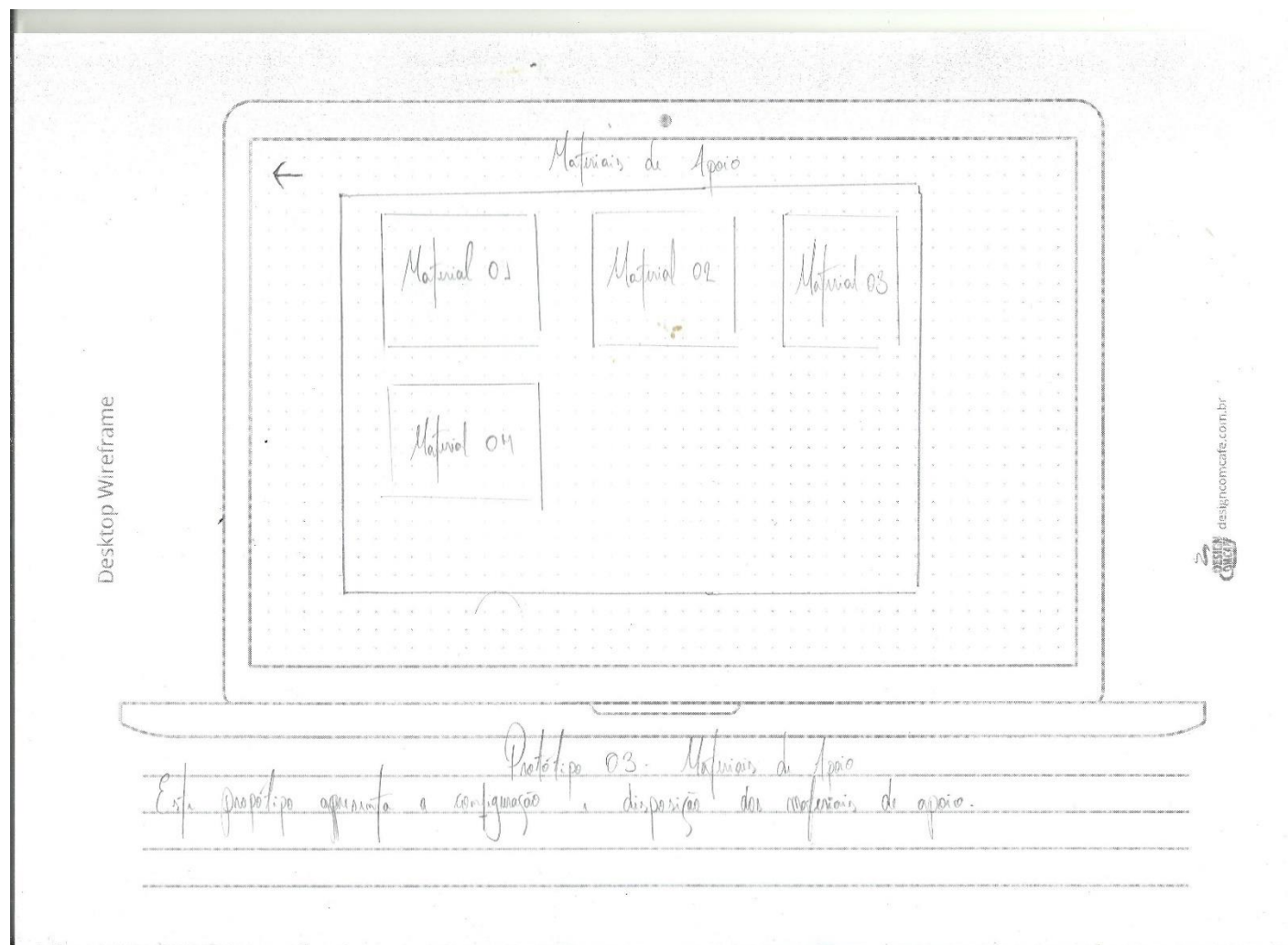
Fonte: Autoria própria

Figura 12. Protótipo de baixa fidelidade: resolução de exercício



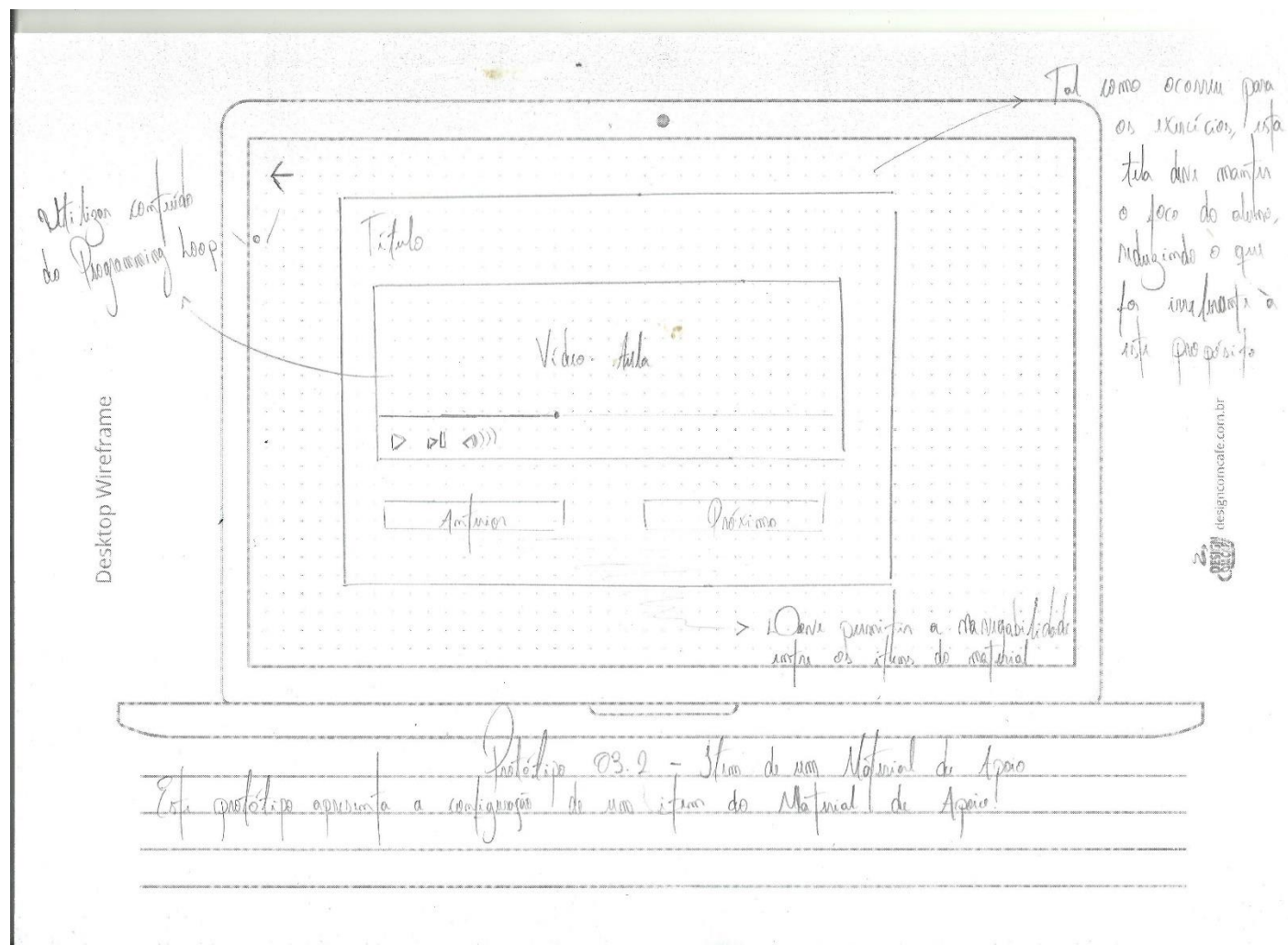
Fonte: Autoria própria

Figura 13. Protótipo de baixa fidelidade: materiais de apoio



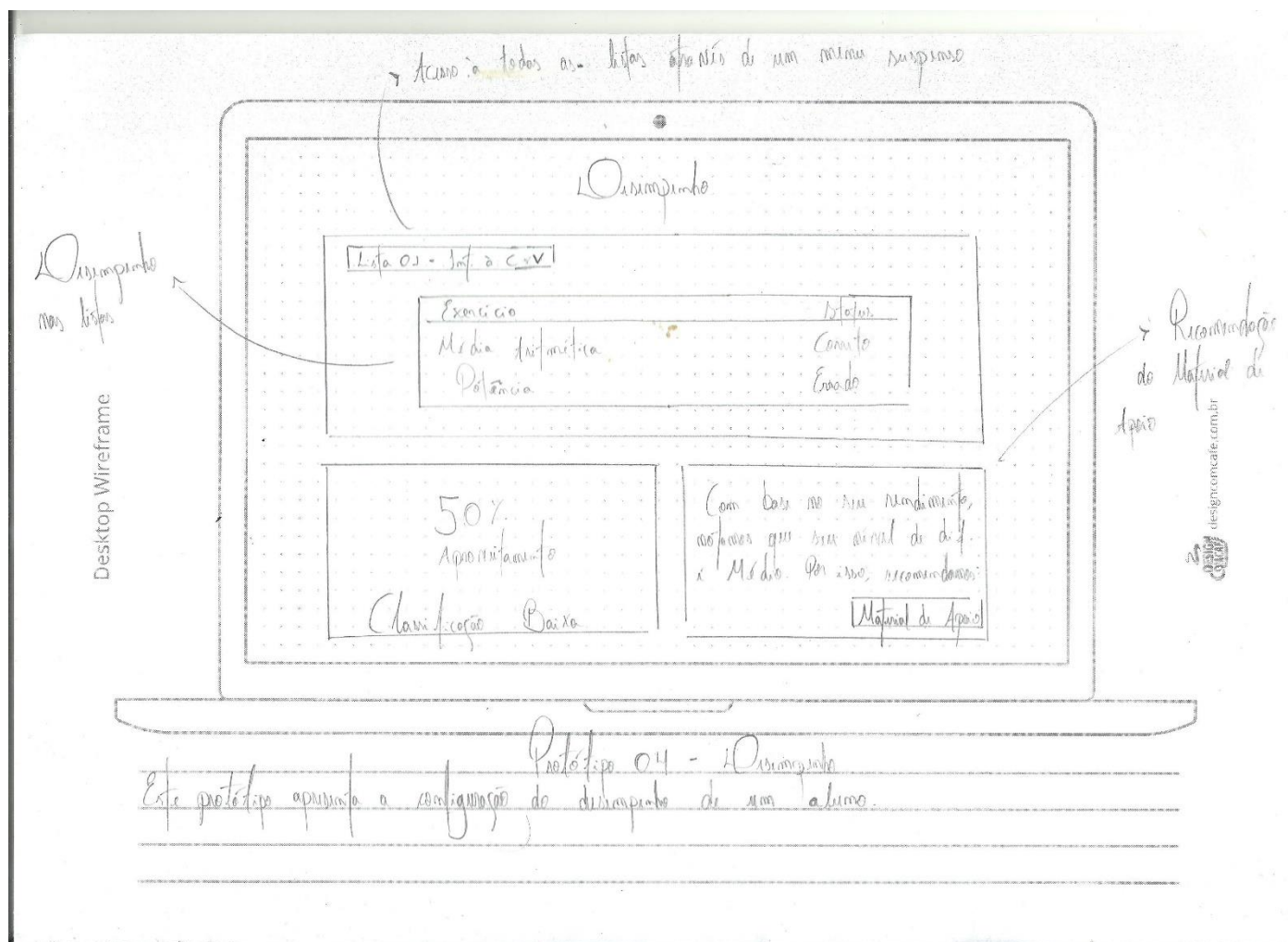
Fonte: Autoria própria

Figura 14. Protótipo de baixa fidelidade: estudo de um item do material de apoio



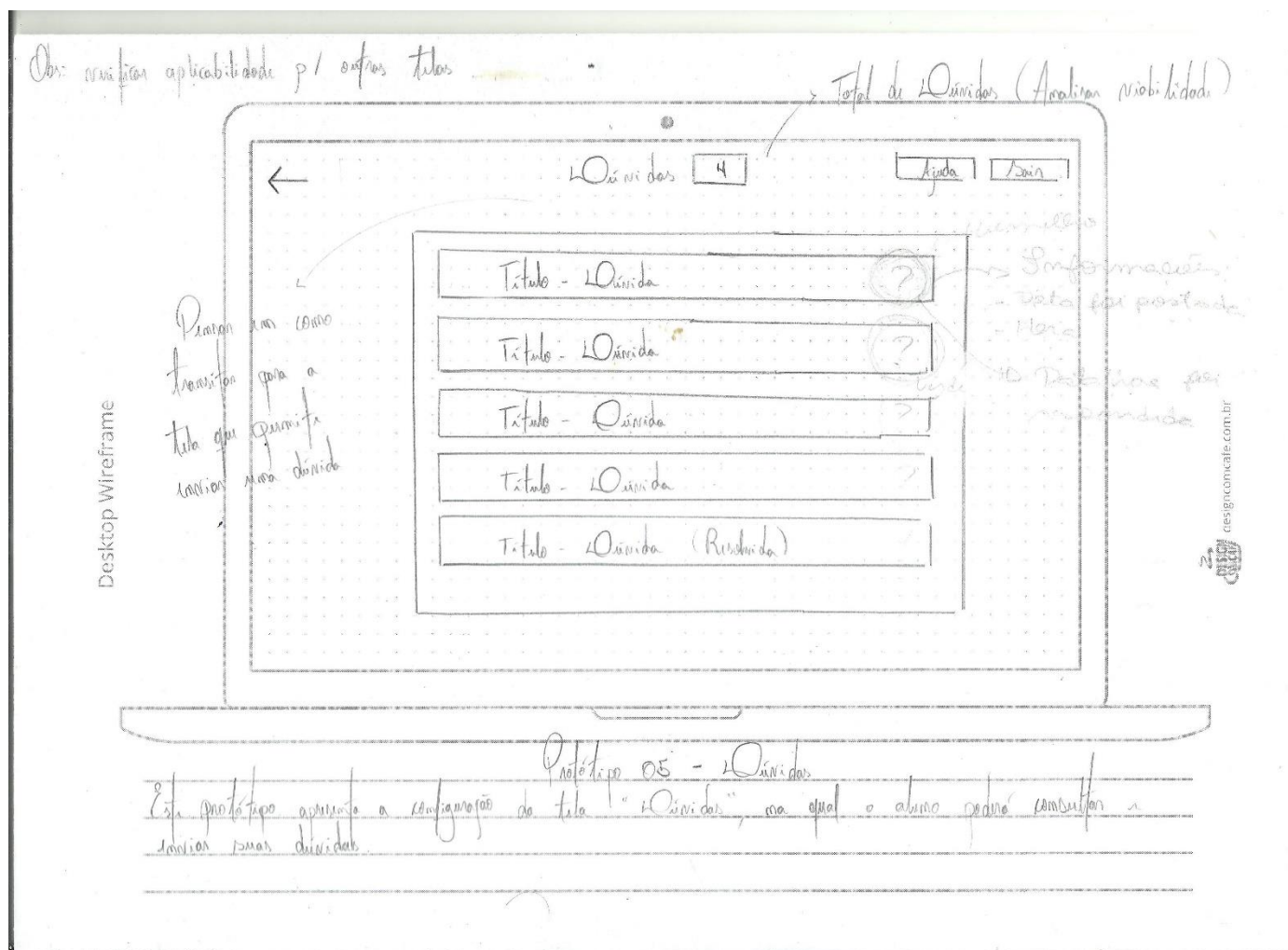
Fonte: Autoria própria.

Figura 15. Protótipo de baixa fidelidade: desempenho



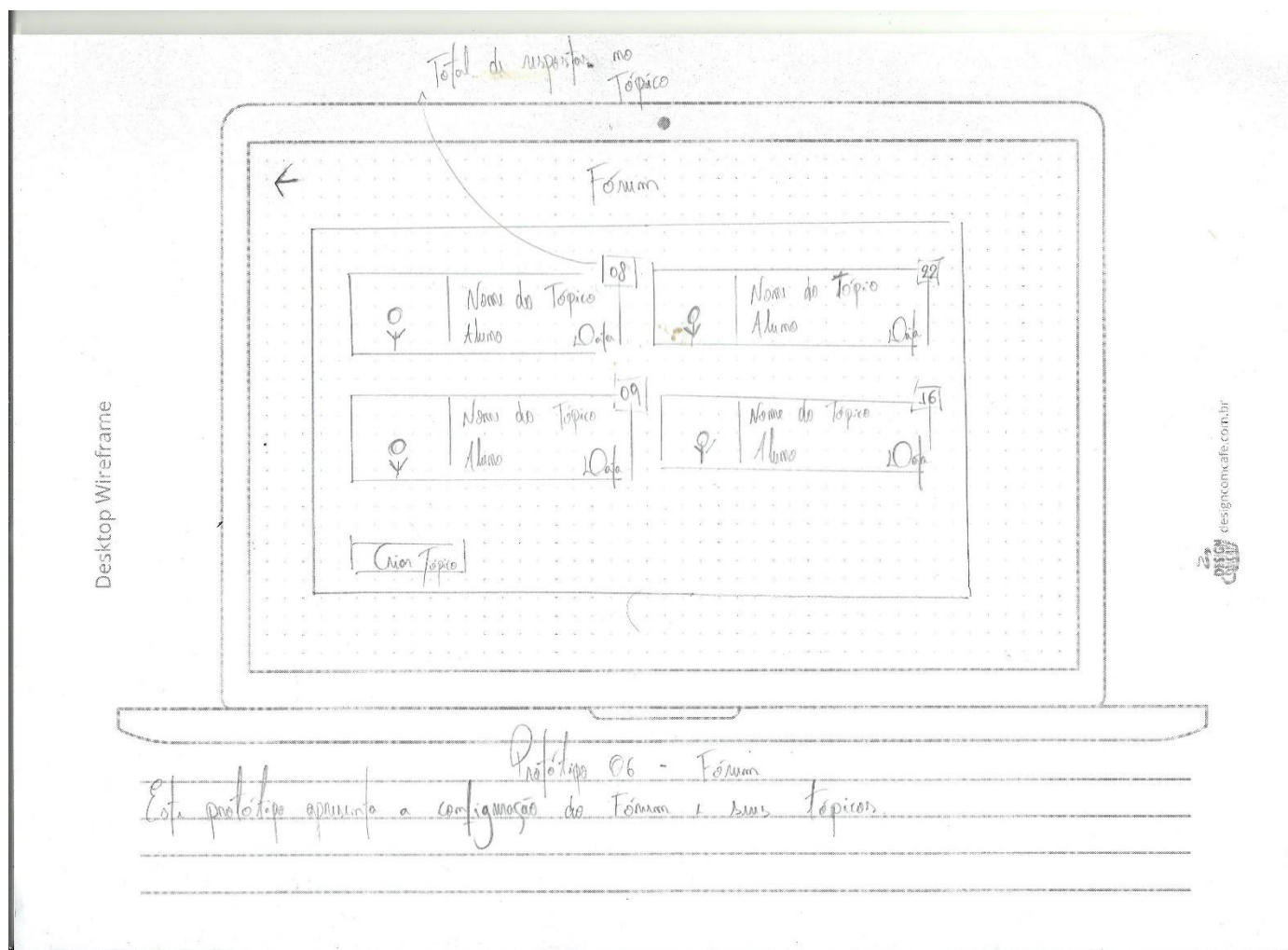
Fonte: Autoria própria.

Figura 16. Protótipo de baixa fidelidade: dúvidas



Fonte: Autoria própria.

Figura 17. Protótipo de baixa fidelidade: fórum



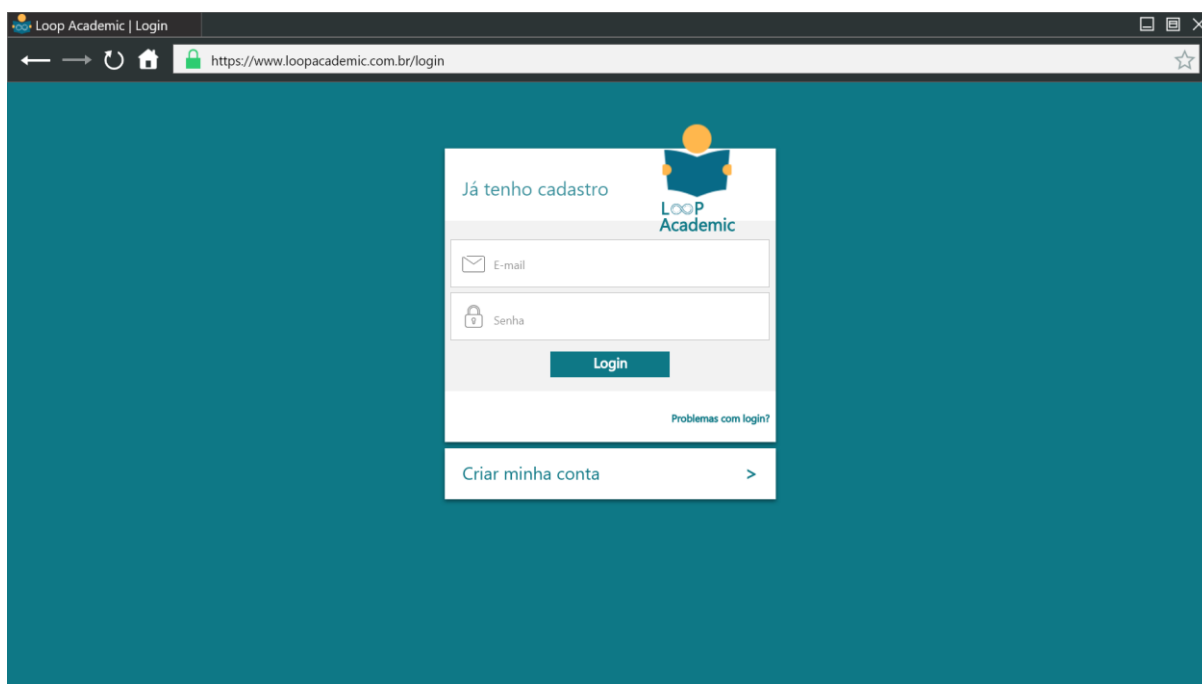
Fonte: Autoria própria.

4.4.3 Telas do Protótipo de Alta Fidelidade

A concepção das telas do protótipo de alta fidelidade do *Loop Academic* ocorreu tomando como base os protótipos de baixa fidelidade apresentados na Subseção 4.4.2.

Apresenta-se, na Figura 18, a tela de login do *Loop Academic*, que define o marco inicial na interação com as funcionalidades propostas. Nela, o usuário tem a opção de realizar login, caso já possua as credenciais de acesso, ou criar a sua conta. Ao clicar em “Criar minha conta”, o usuário é direcionado à Figura 19, onde deve informar se é Aluno, Professor ou Monitor. Ao informar ser aluno, o usuário é direcionado à Figura 20, tela na qual deve informar seus dados cadastrais.

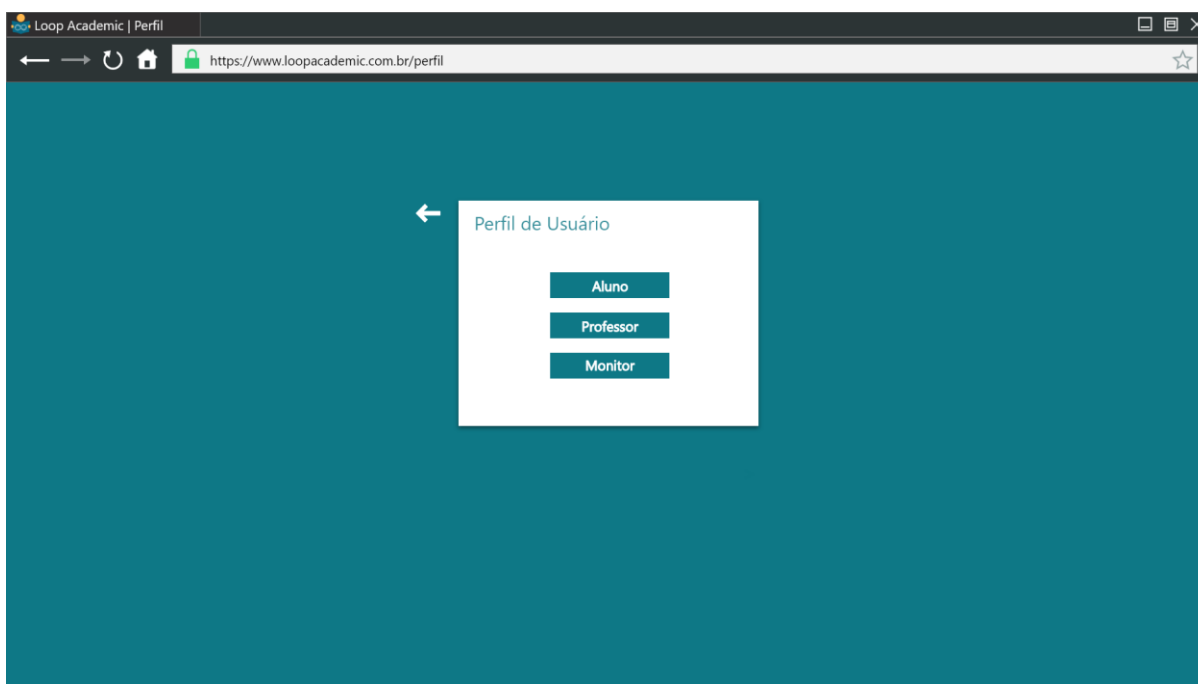
Figura 18. Protótipo de alta fidelidade: login



Fonte: Autoria própria

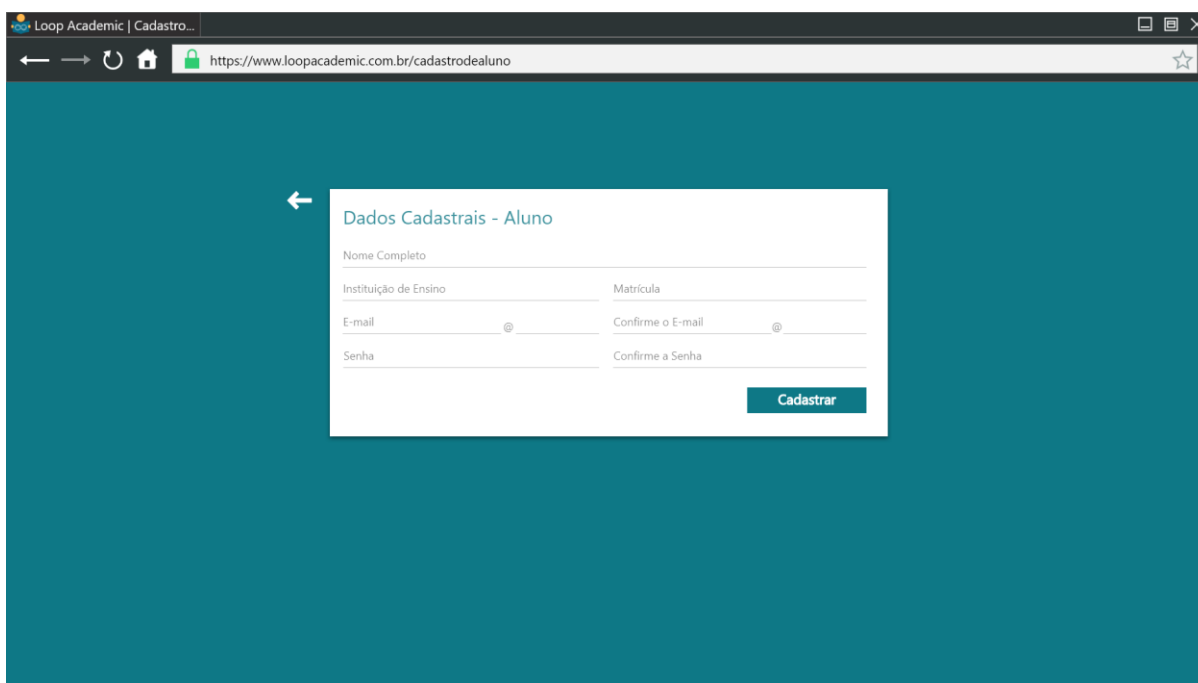
Após a realização do seu cadastro e login no *Loop Academic*, o aluno irá deparar-se com uma tela solicitando o código da turma virtual, conforme demonstra a Figura 21. Essa estratégia foi adotada para garantir que os alunos de uma determinada turma tenham acesso aos materiais direcionados à sua turma, a citar: listas de exercícios e apostilas. Ressalta-se que o código da turma virtual somente pode ser obtido pelo professor, que ficará responsável por criar e gerenciar a turma virtual e ofertar aos alunos o código da turma virtual.

Figura 19. Protótipo de alta fidelidade: perfil



Fonte: Autoria própria

Figura 20. Protótipo de alta fidelidade: cadastro



Fonte: Autoria própria

Figura 21. Código da Turma Virtual

The screenshot shows a web browser window with the address bar displaying "https://www.loopacademic.com.br/home". The page has a teal header and footer. A central white box contains the following text:

Boas vindas!

Olá, futuro(a) programador(a)

É com enorme satisfação que desejamos-lhes nossas boas vindas ao Loop Academic! \o/

Neste Software Educacional, você terá acesso à uma serie de funcionalidades que lhe auxiliarão no aprendizado de Algoritmos, por exemplo: listas de exercícios de programação, material de apoio ao estudo de Algoritmos, envio de dúvidas ao professor ou monitor, fórum de interação com os colegas do curso e muito mais! o/

Para ter acesso à todas as funcionalidades, é necessário que você informe abaixo o Código da Turma Virtual criada pelo seu professor. É com ele que você terá acesso aos materiais exclusivos da sua turma!

Por favor, digite o seu Código da Turma Virtual

Código da Turma Virtual

At the bottom right of the form are two buttons: "Não Tenho" (red) and "Enviar" (green).

Fonte: Autoria própria

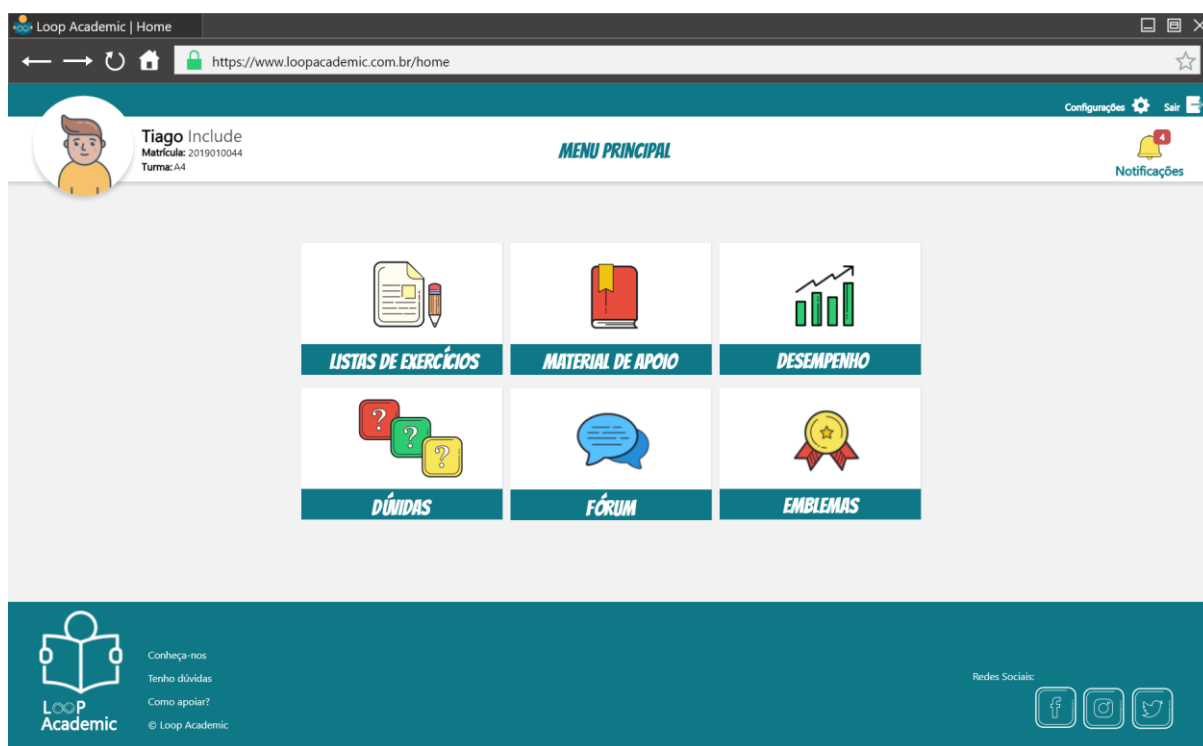
Após concluir as etapas direcionadas ao seu cadastro e login, o aluno é direcionado à tela principal do protótipo, que representa o menu principal, local onde as funcionalidades do *Loop Academic* têm seu início, conforme ilustrado na Figura 22. De acordo com o que já foi descrito nos requisitos funcionais do *Loop Academic* (Subseção 4.2), o aluno tem acesso às seguintes funcionalidades: (i) listas de exercícios; (ii) materiais de apoio; (iii) desempenho; (iv) dúvidas; (v) fórum; (vi) emblemas. Cada uma dessas funcionalidades principais será discutida nas próximas subseções.

Analisando o Menu Principal (ver Figura 22), nota-se que no canto superior direito há um ícone que indica notificações. A presença desse ícone tem como principal direcionamento alertar o aluno acerca de atualizações relevantes no *Loop Academic*, tais como: nova lista de exercícios, novo material de estudo, dúvidas respondidas, emblemas obtidos e outros aspectos interessantes.

Outra observação relevante e necessária consiste em enfatizar que o perfil do Tiago Include, persona representativo do aluno novato (ver Subseção 4.3.1), é utilizado. A escolha por representar o seu perfil têm como objetivo demonstrar as formas nas quais as funcionalidades do *Loop Academic* foram desenvolvidas, i.e., o que foi projetado de forma a ajudá-lo na resolução dos problemas listados no seu perfil de persona. Também ressalta-se a

participação da Tereza Return, persona representativo da aluna veterana (vide Subseção 4.3.2.), em um outro momento.

Figura 22. Protótipo de alta fidelidade: menu principal



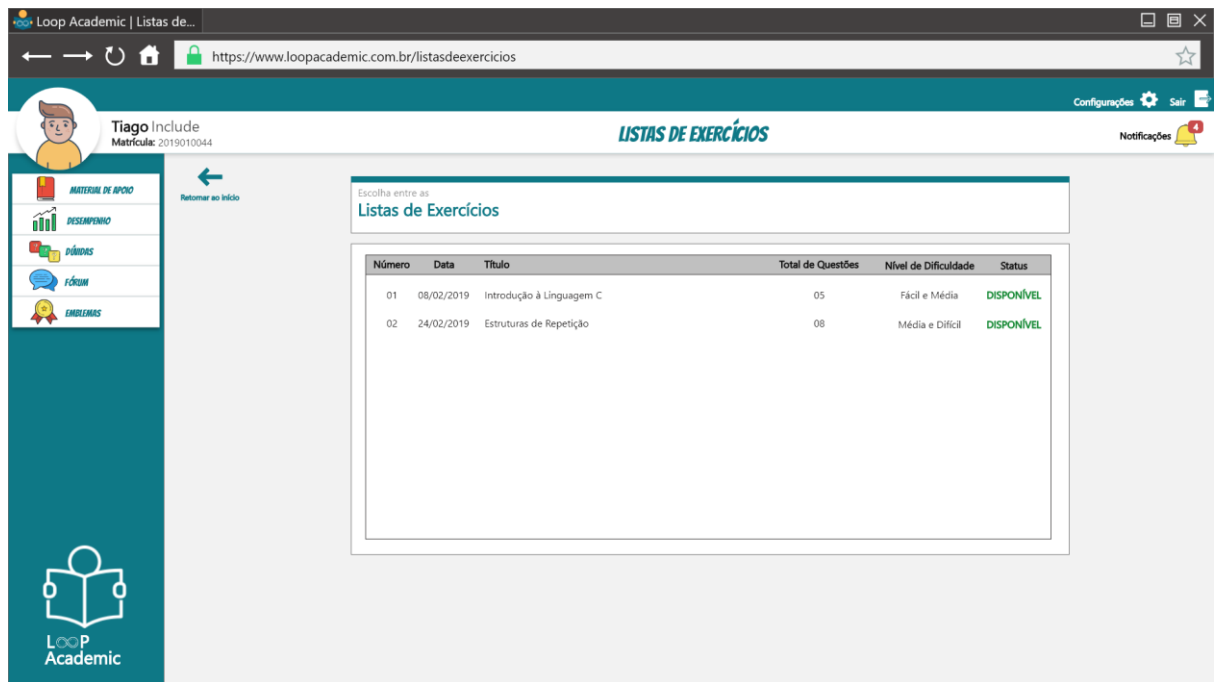
Fonte: Autoria própria

4.4.2.1 Listas de Exercícios

O acesso ao item “Listas de Exercícios” direciona o aluno à tela representada na Figura 23, na qual o aluno, ao escolher uma das listas disponibilizadas, é redirecionado à tela da Figura 24, que consiste nos exercícios que compõem a lista de exercícios.

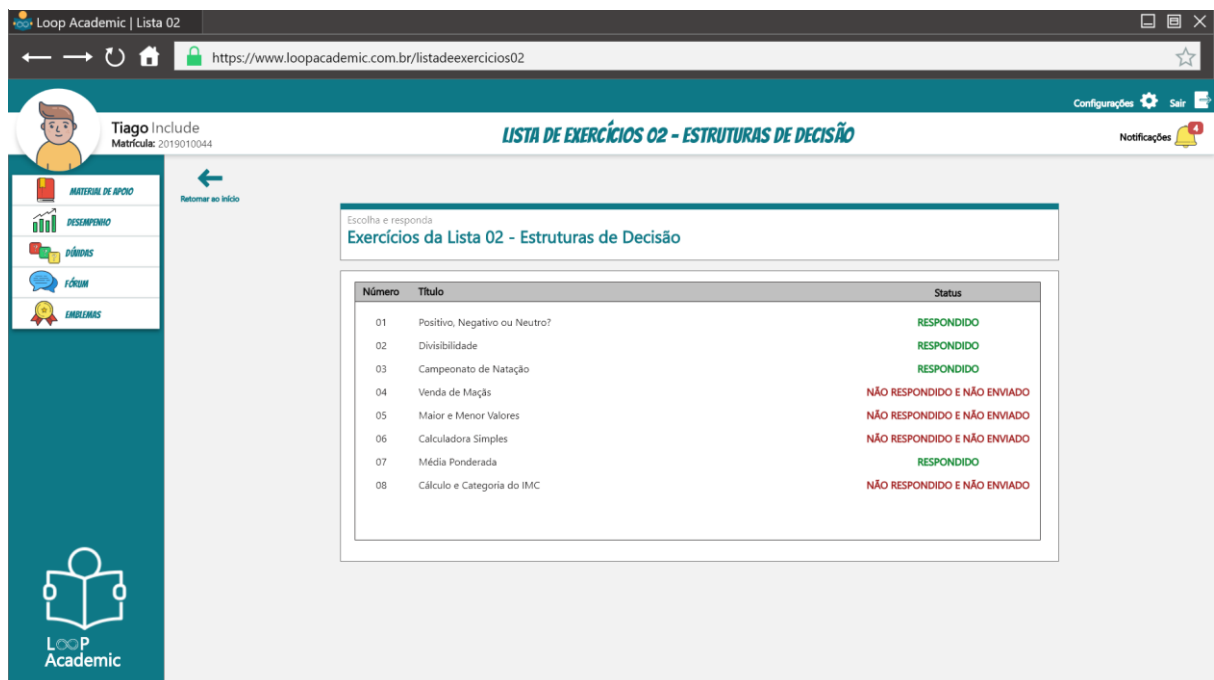
Com a execução dos passos anteriormente descritos e a escolha de um exercício, o aluno pode resolver um exercício em uma tela direcionada ao foco (ver Figura 25), na qual as informações gerais do *Loop Academic* foram retiradas, deixando somente os mecanismos para: (i) resolver o exercício, como é o caso da área apresentada na cor preta, que representa uma IDE; (ii) acesso de dicas, código de apoio e material de apoio, que podem ser utilizados para sanar suas dúvidas; (iii) menu navegável à direita, que consiste na representação de todos os exercícios da lista de exercícios, de forma a facilitar a navegabilidade.

Figura 24. Protótipo de alta fidelidade: listas de exercícios



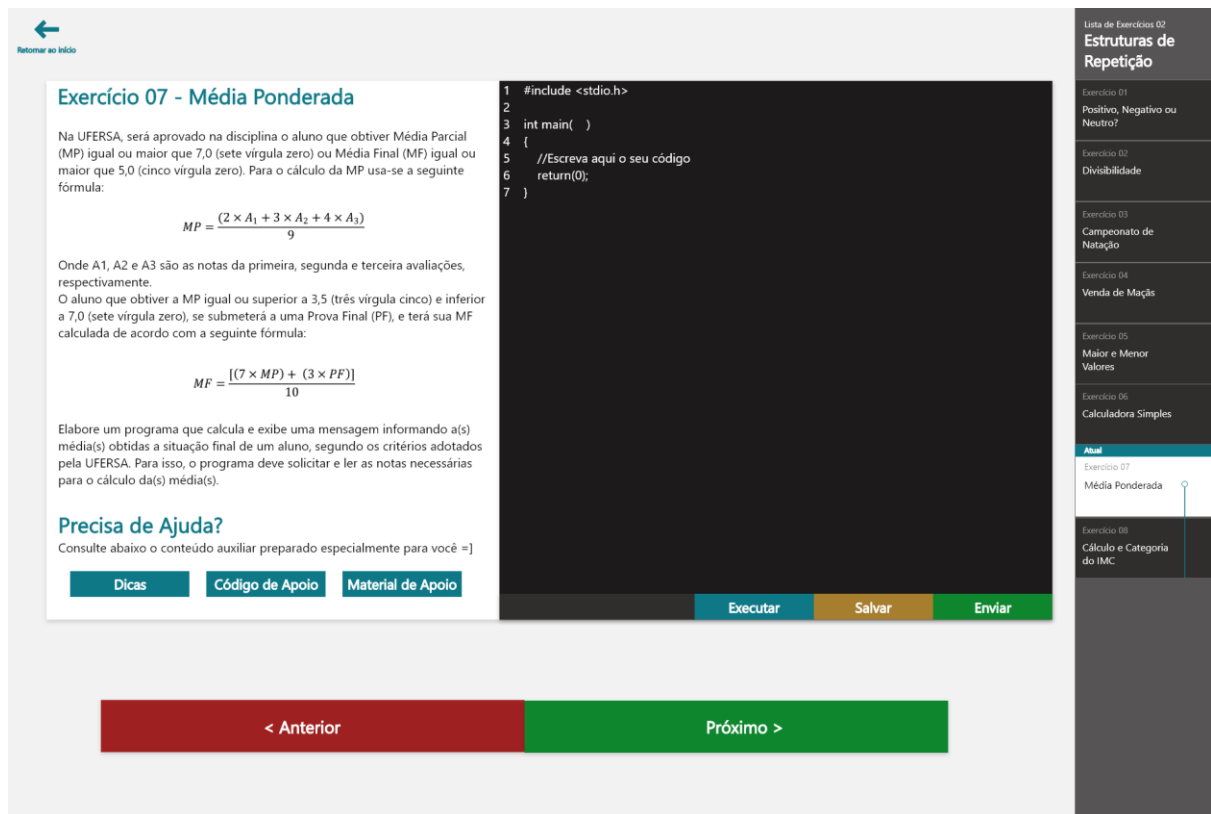
Fonte: Autoria própria

Figura 23. Protótipo de alta fidelidade: exercícios de uma lista de exercício



Fonte: Autoria própria

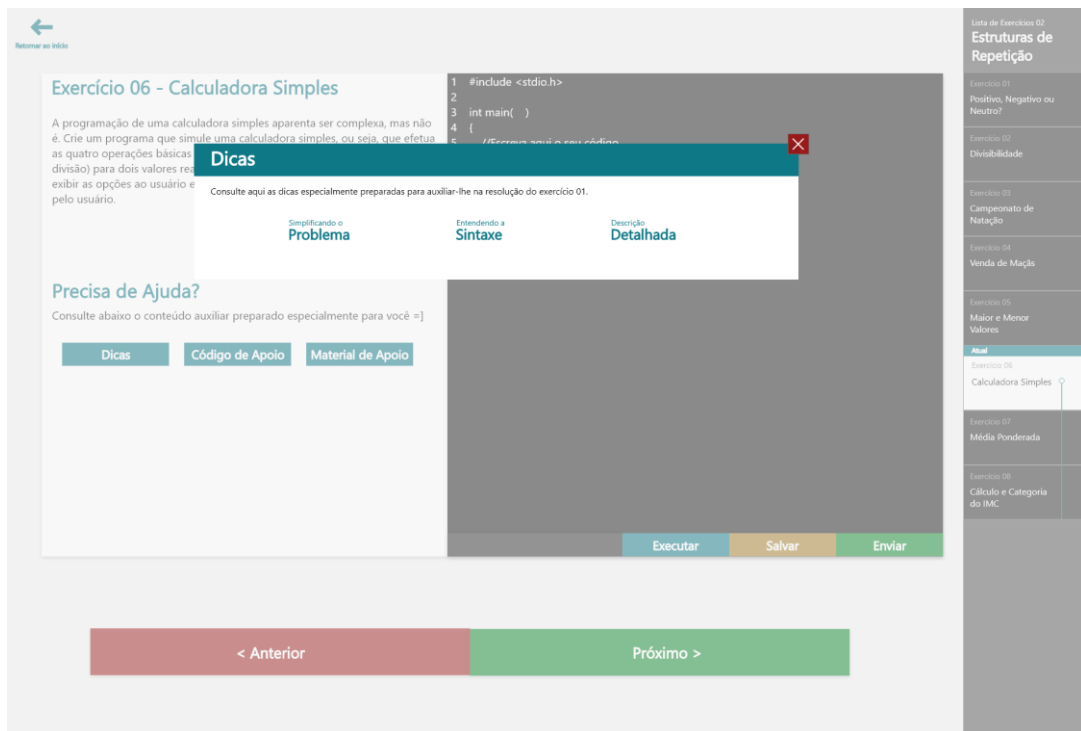
Figura 25. Protótipo de alta fidelidade: exercícios de uma lista de exercícios



Fonte: Autoria própria

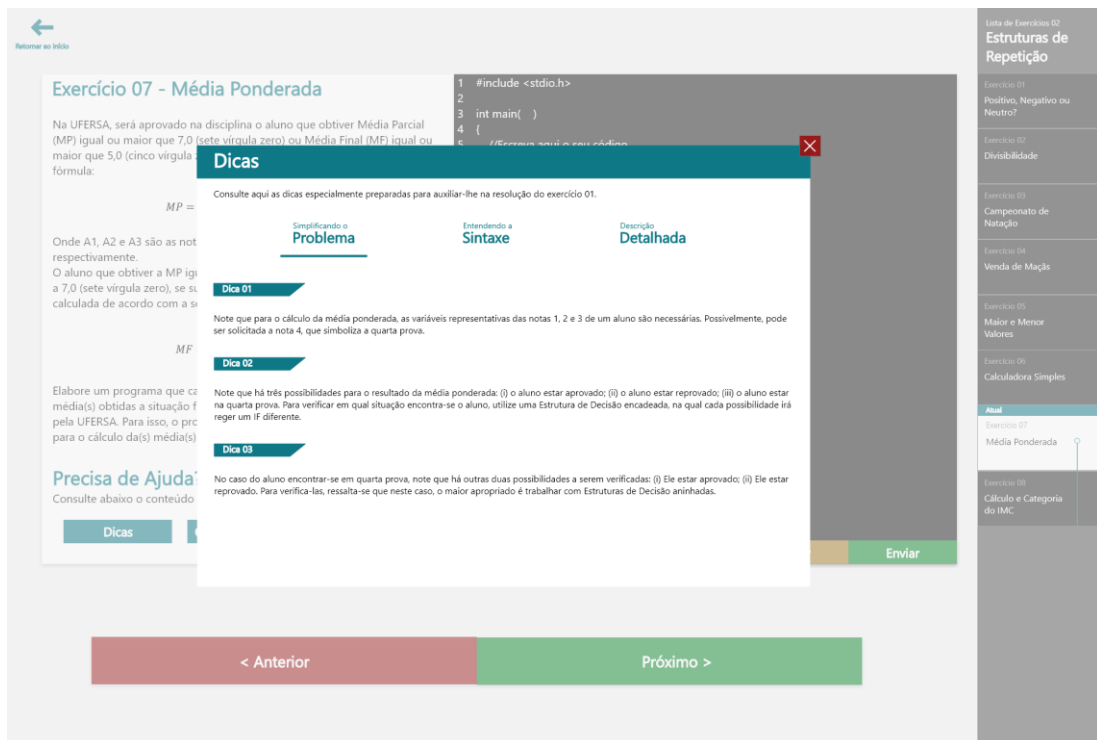
Ao clicar no botão “Dicas” ou “Código de Apoio”, surge uma janela na tela de resolução de exercício com a informação solicitada. Enfatiza-se que as dicas são classificadas em três categorias: (i) Simplificando o Problema; (ii) Entendendo a Sintaxe; (iii) Descrição Detalhada. Cada classificação envolve um grau de abstração e detalhismo diferente, sendo a primeira marcada por um alto nível de abstração e a terceira marcada por um maior detalhismo, conforme ilustrado nas Figuras 26, 27 e 28. A dica “Entendendo a Sintaxe” apresenta uma revisão sobre a sintaxe dos principais comandos que o aluno irá utilizar na resolução do seu exercício, i. e., se ele está resolvendo exercícios de Estruturas de Decisão, consequentemente, lhe será disponibilizada uma tela com a sintaxe das estruturas IF, IF-ELSE, IF-ELSE-IF, SWITCH, etc. O Código de Apoio consiste na apresentação de um algoritmo já resolvido e com temática similar ao exercício.

Figura 26. Protótipo de alta fidelidade: dicas de um exercício



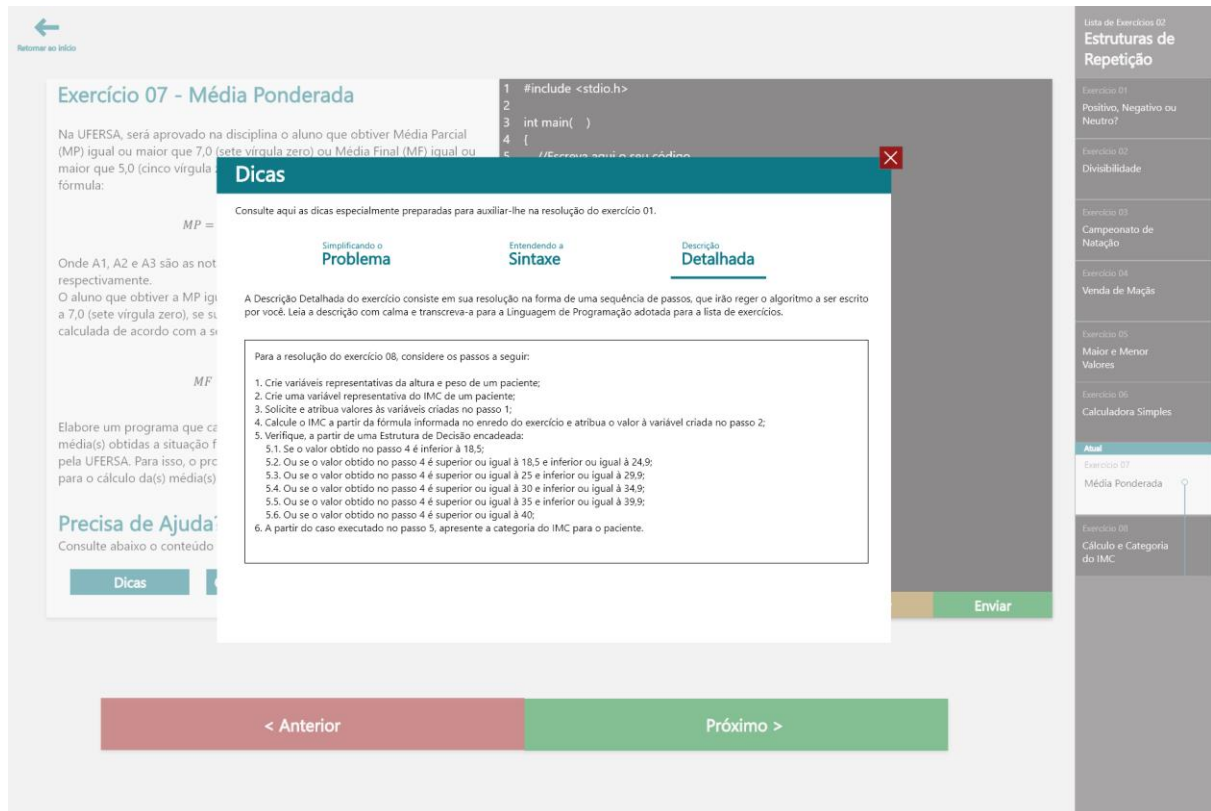
Fonte: Autoria própria

Figura 27. Protótipo de alta fidelidade: simplificando o problema



Fonte: Autoria própria

Figura 28. Protótipo de alta fidelidade: descrição detalhada



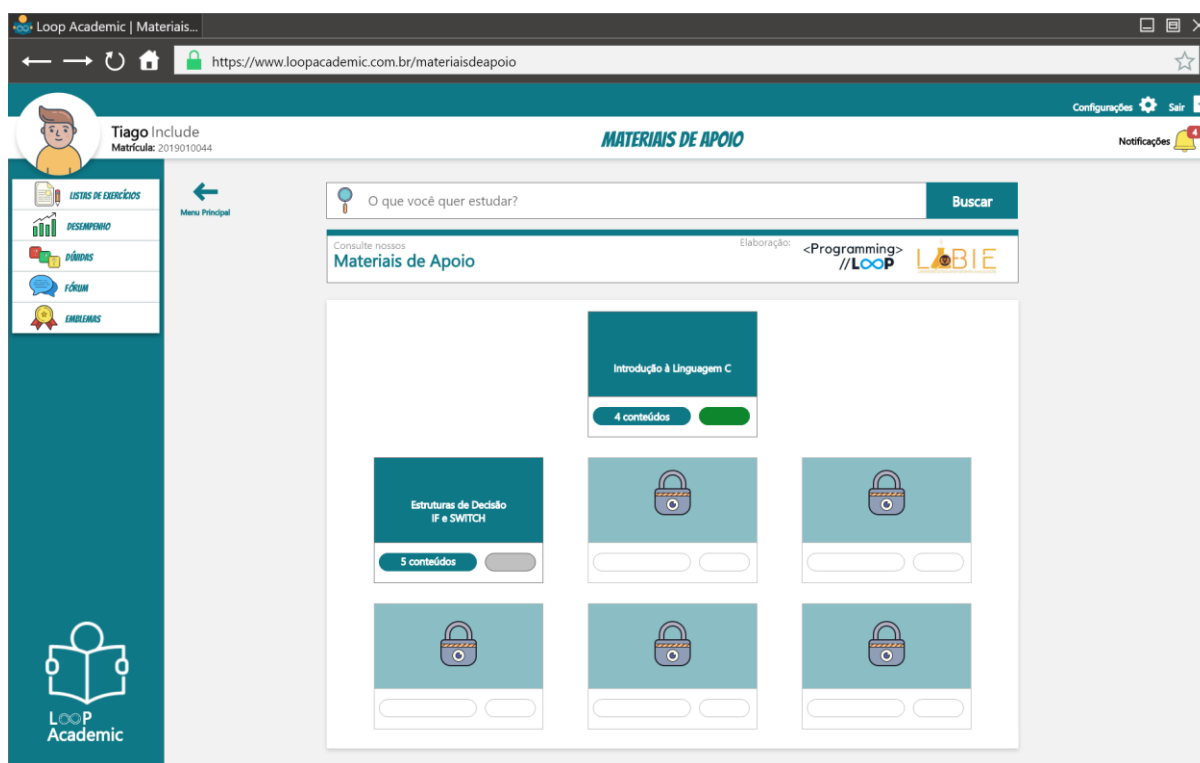
Fonte: Autoria própria

4.4.2.2 Material de Apoio

O acesso ao item “Material de Apoio” direciona o aluno às telas 29 e 30, que consistem, respectivamente, na escolha de um material de apoio e na escolha de um item do material de apoio escolhido. Enfatiza-se que a elaboração desses artefatos estará encarregada ao LABIE e ao canal *Programming Loop*.

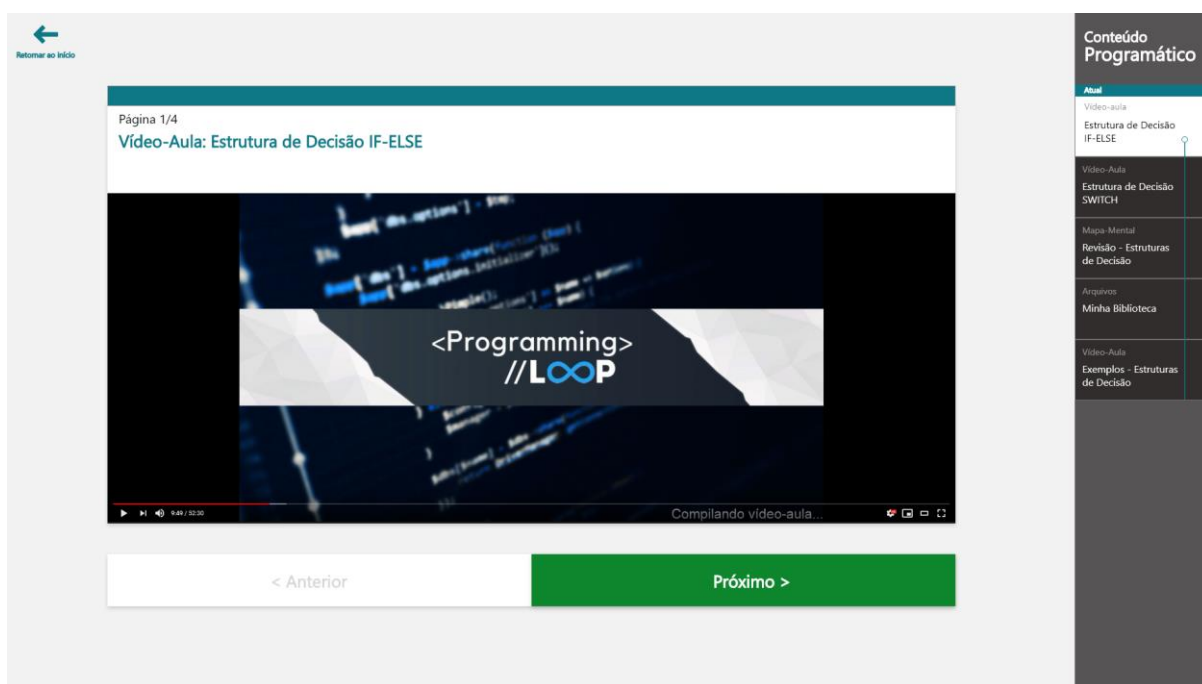
Com a escolha de um item, o estudo é realizado em um ambiente direcionado ao foco, tal qual ocorre com a resolução de um exercício. Neste ambiente, há um menu vertical à direita, que se encarrega de apresentar todos os itens que compõem o respectivo material de apoio. A sua tela é apresentada na Figura 31.

Figura 29. Protótipo de alta fidelidade: materiais de apoio



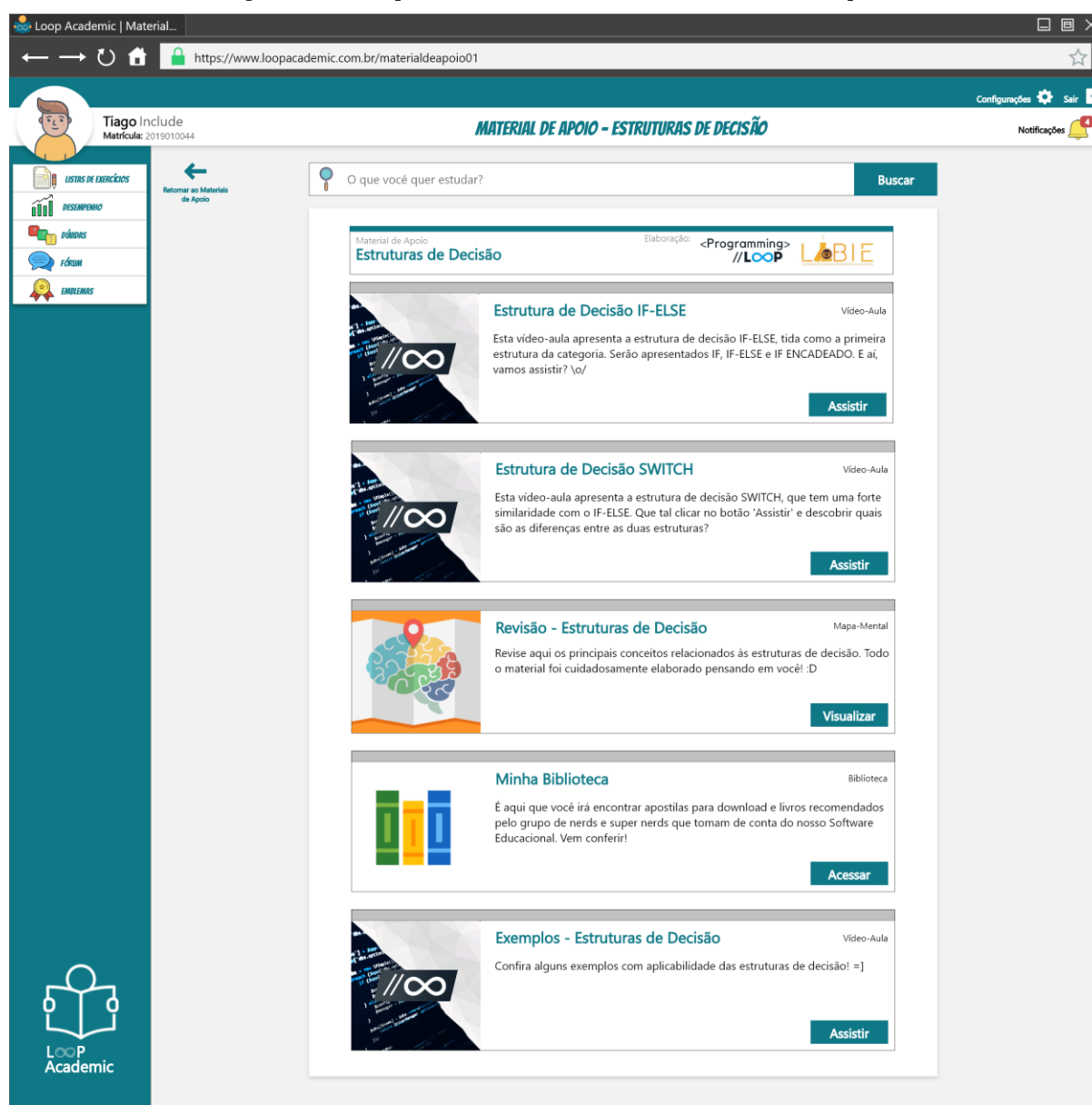
Fonte: Autoria própria

Figura 30. Protótipo de alta fidelidade: itens de um material de apoio



Fonte: Autoria própria

Figura 31. Protótipo de alta fidelidade: itens de um material de apoio

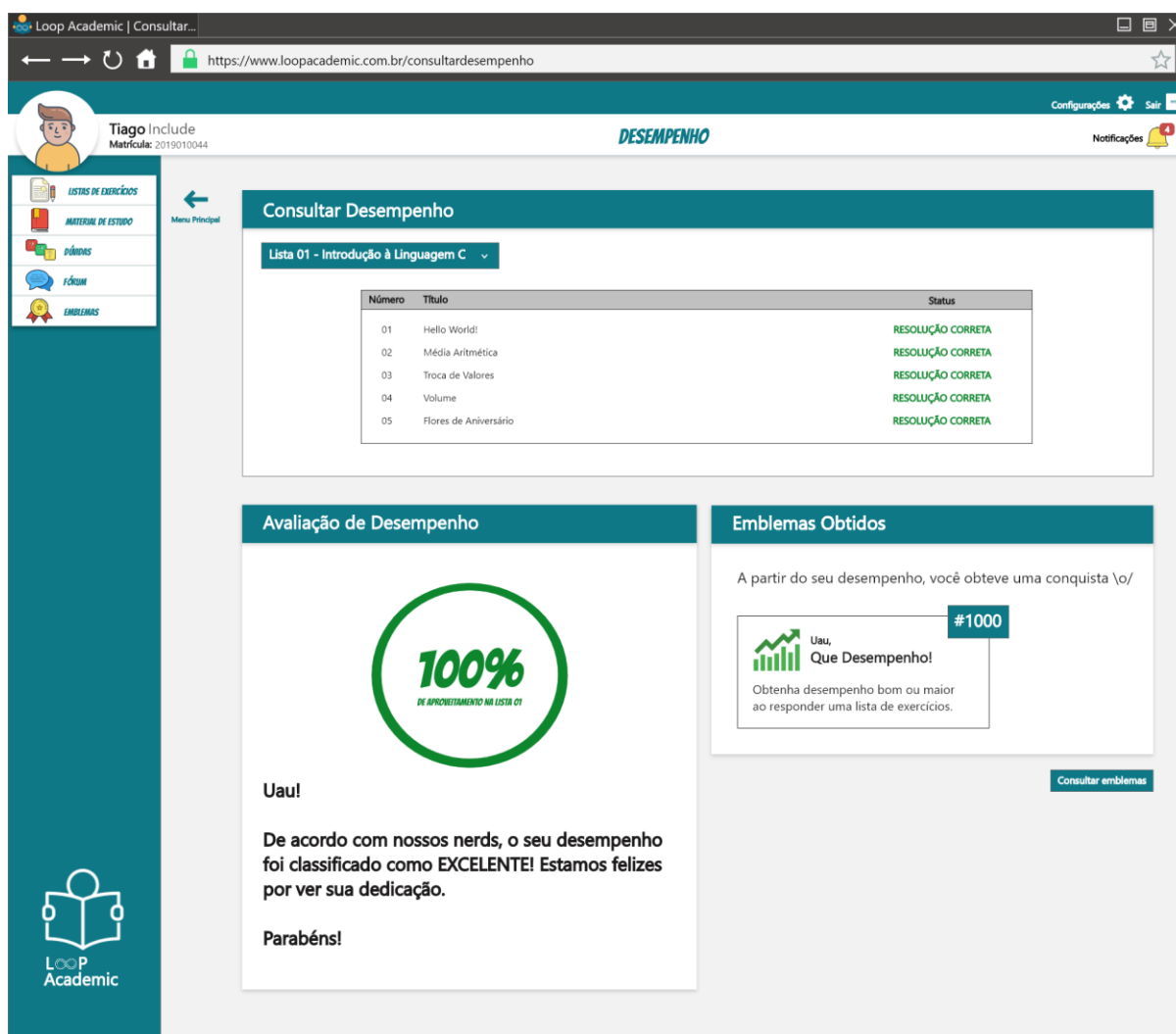


Fonte: Autoria própria

4.4.2.3 Desempenho

O acesso ao item “Desempenho” direciona o aluno à interface mostrada na Figura 32, que apresenta o seu desempenho na resolução das listas de exercícios disponibilizadas. O intuito dessa tela é apresentar ao aluno os seus níveis de rendimentos. Para estimulá-lo, há emblemas que são desbloqueados à medida que bons rendimentos nas listas são alcançados - os emblemas encontram-se na Figura 40.

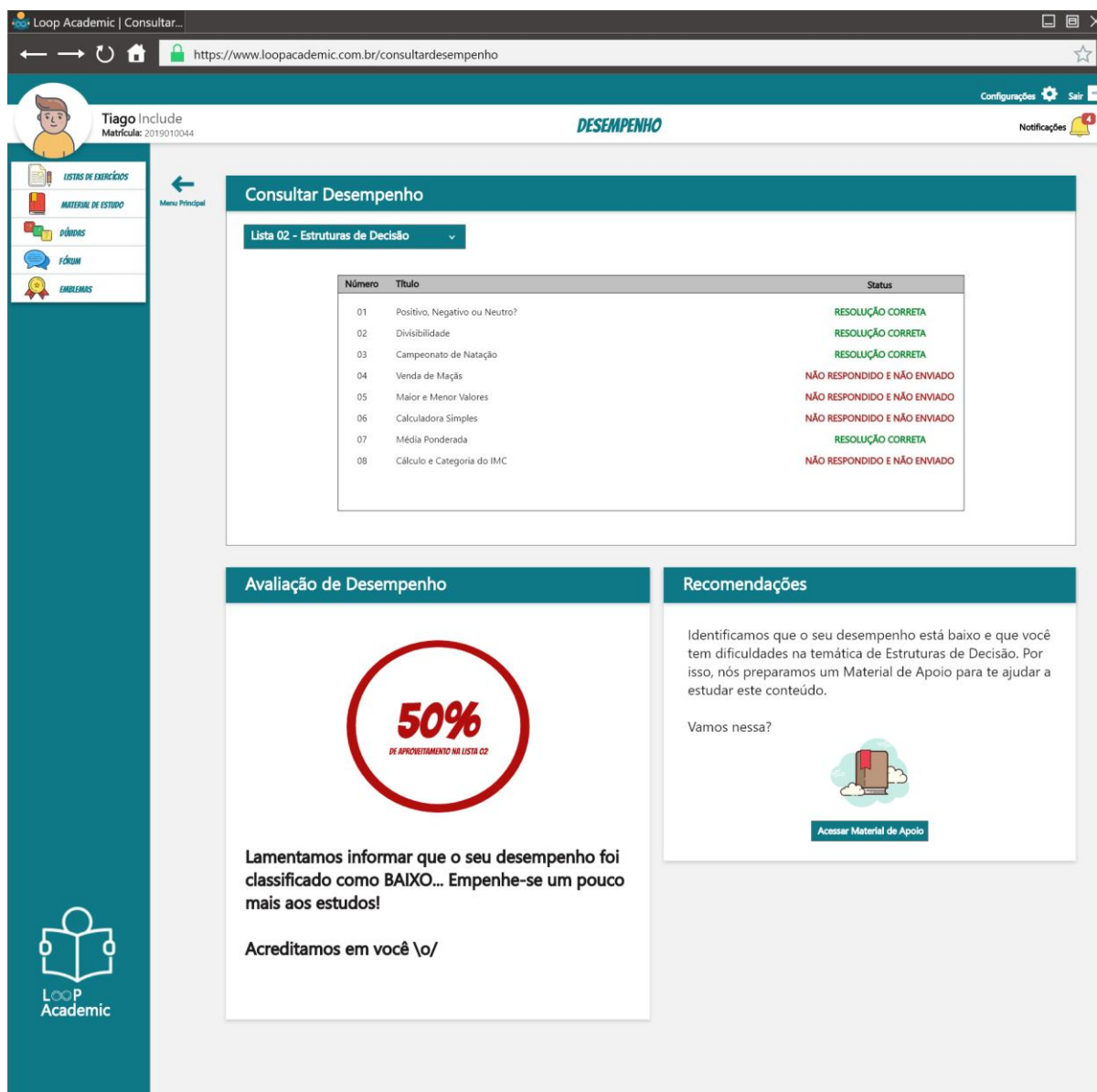
Figura 32. Protótipo de alta fidelidade: excelente desempenho



Fonte: Autoria própria, 2019

Também é válido ressaltar que na ocorrência de baixos desempenhos, o *Loop Academic* sugere o estudo de materiais de apoio com a mesma temática de dificuldade, i.e., se o aluno obter um baixo desempenho na resolução de exercícios sobre Estruturas de Decisão, é sugerido o estudo dos materiais de apoio sobre Estruturas de Decisão. Isso é demonstrado na Figura 33, na qual o Tiago Include obteve uma taxa de aproveitamento da Lista 02 - Estruturas de Decisão em torno de 50%. O *Loop Academic* classificou o seu desempenho como baixo e direcionou-lhe um material de estudo.

Figura 33. Protótipo de alta fidelidade: baixo desempenho

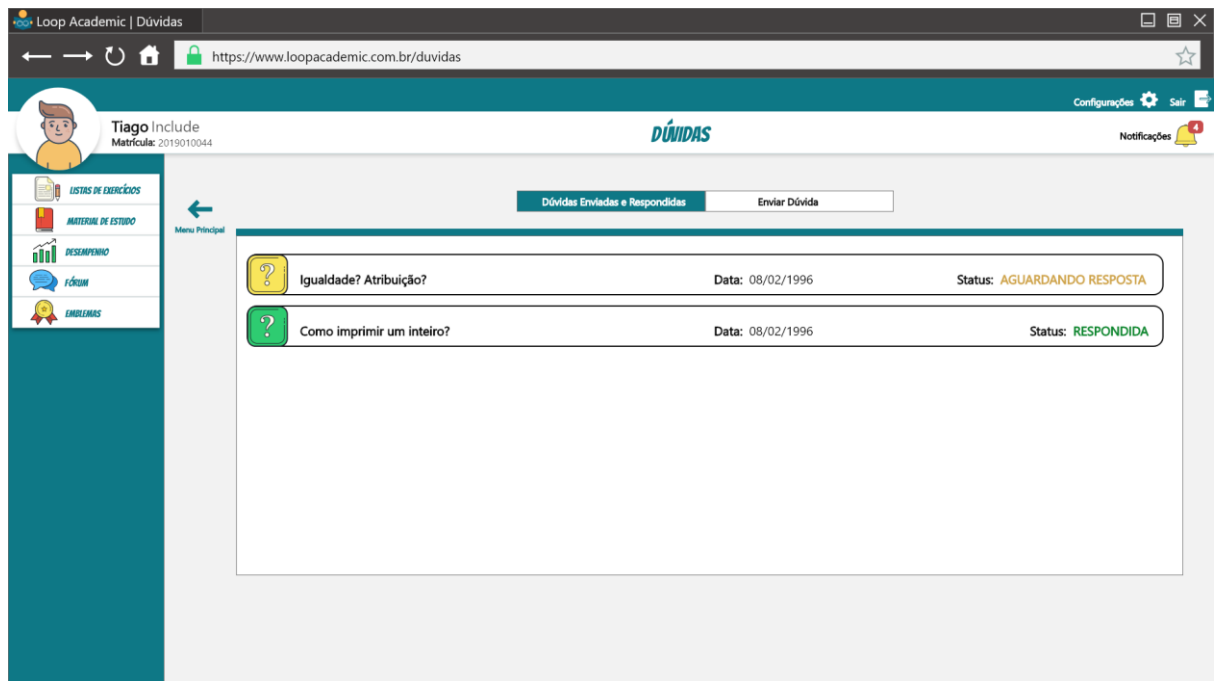


Fonte: Autoria própria, 2019

4.4.2.4 Dúvidas

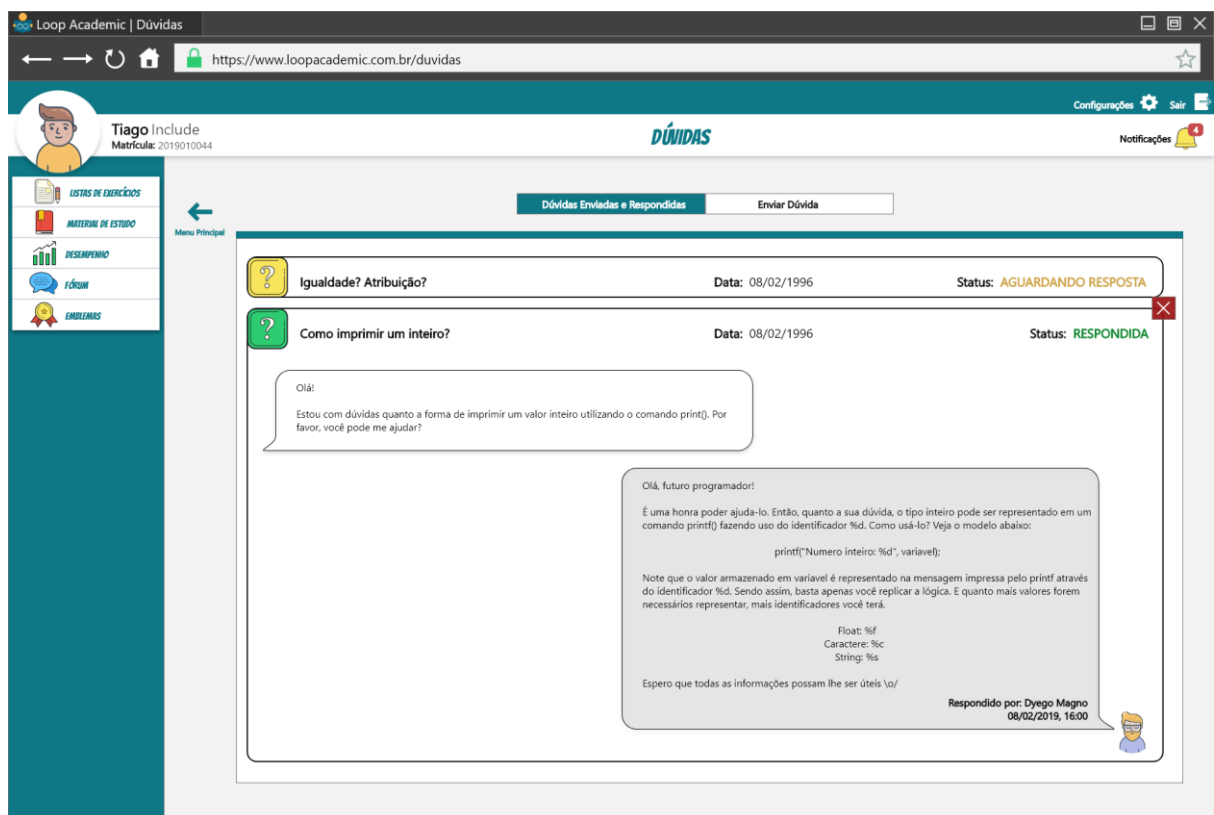
O acesso ao item “Dúvidas” direciona o aluno à tela 34, na qual ele pode acessar as dúvidas enviadas e respondidas ou enviar uma nova dúvida ao professor. Este é um recurso que visa auxiliar o aluno que possui algum tipo de receio em tirar suas dúvidas com o professor na sala de aula ou nos horários de atendimento. Ressalta-se que como o *Loop Academic* visa integrar o trabalho do monitor ao professor, sendo que este poderá auxiliar o professor na retirada de dúvidas, tal qual ocorre na Figura 35.

Figura 34. Protótipo de alta fidelidade: dúvidas



Fonte: Autoria própria

Figura 35. Protótipo de alta fidelidade: dúvida respondida



Fonte: Autoria própria

Em relação ao envio de dúvidas, esse processo será gerado de forma anônima desde que o usuário não opte por identificar-se. A preferência pelo anonimato tem como premissa principal combater a inibição que os alunos sentem ao retirar suas dúvidas, o que é decorrente da visão de estar perguntando algo muito óbvio ou de deixar evidente que ainda não compreendeu o conteúdo. A Figura 36 demonstra como é realizado o processo de envio de dúvidas ao professor.

Figura 36. Protótipo de alta fidelidade: envio de dúvidas

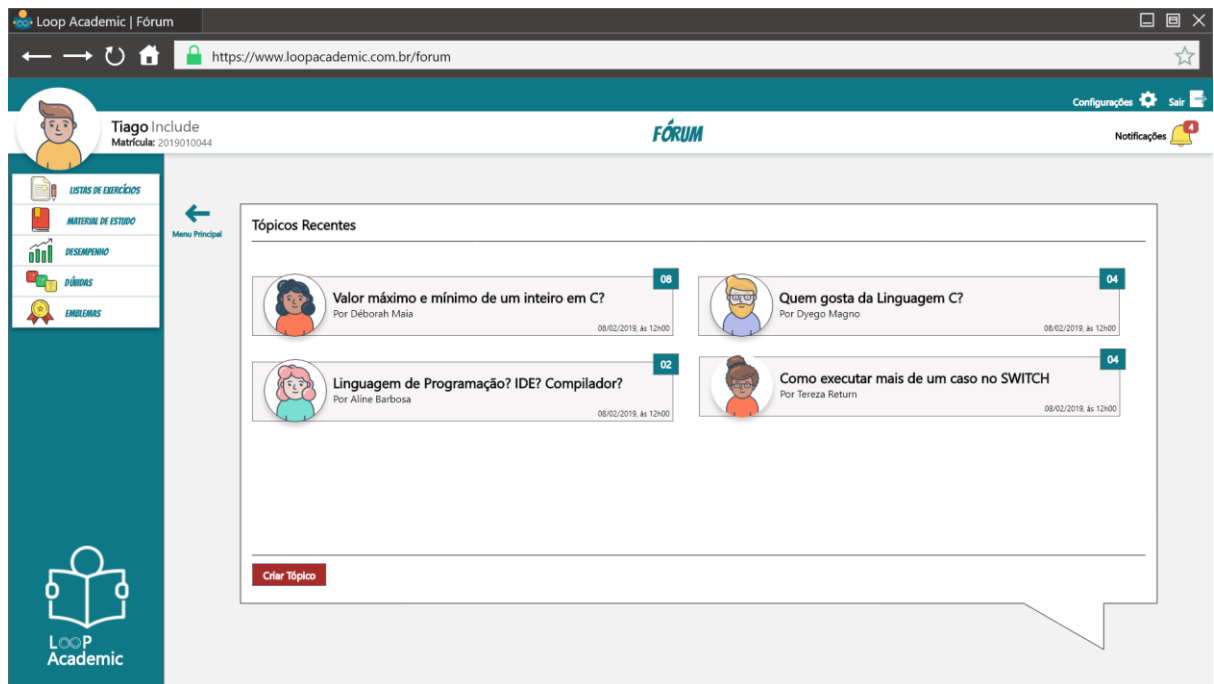
Fonte: Autoria própria, 2019

4.4.2.5 Fórum

O acesso ao item “Fórum” direciona o aluno às telas que caracterizam o fórum, espaço no qual ele pode interagir com outros usuários, seja ajudando-os com suas dúvidas, comentando postagens que julgar relevantes ou até mesmo realizando outras postagens. Em tese, o Fórum é um espaço interativo, que visa unir todos os usuários do *Loop Academic* - alunos, professores e monitores.

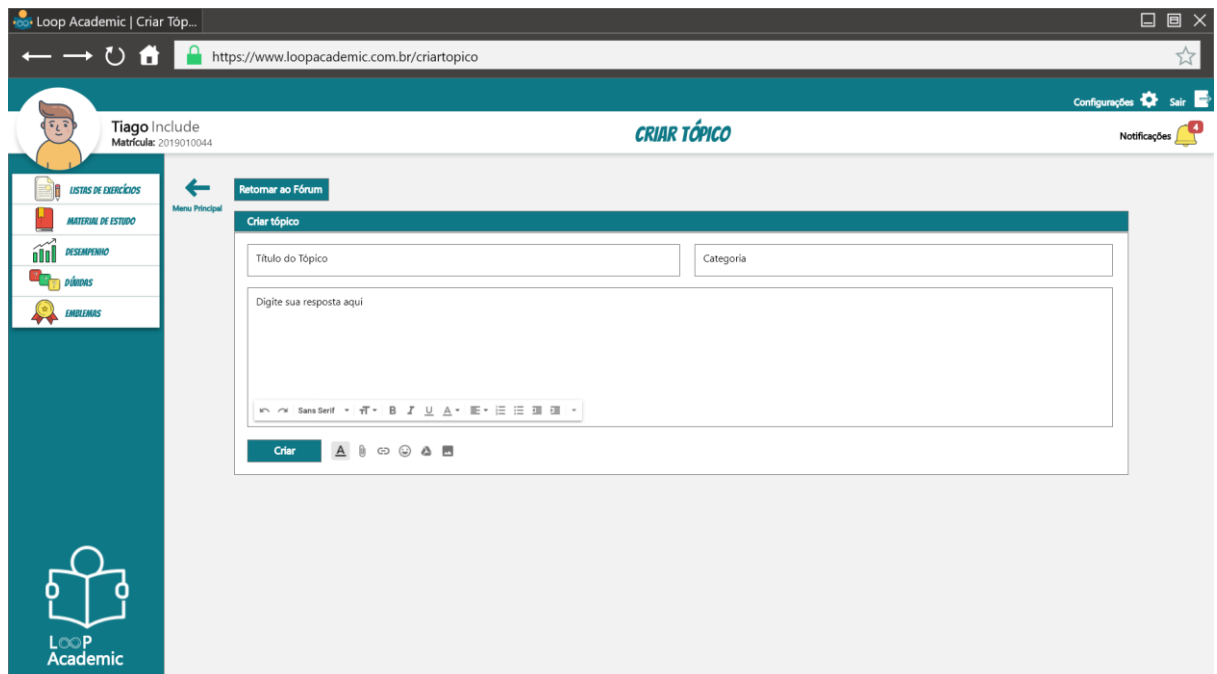
A Figura 37 apresenta a disposição inicial da tela principal do Fórum. A Figura 38 apresenta a página de criação de um tópico no Fórum. E a Figura 39 demonstra como está realizada a disposição das postagens no fórum.

Figura 37. Protótipo de alta fidelidade: fórum



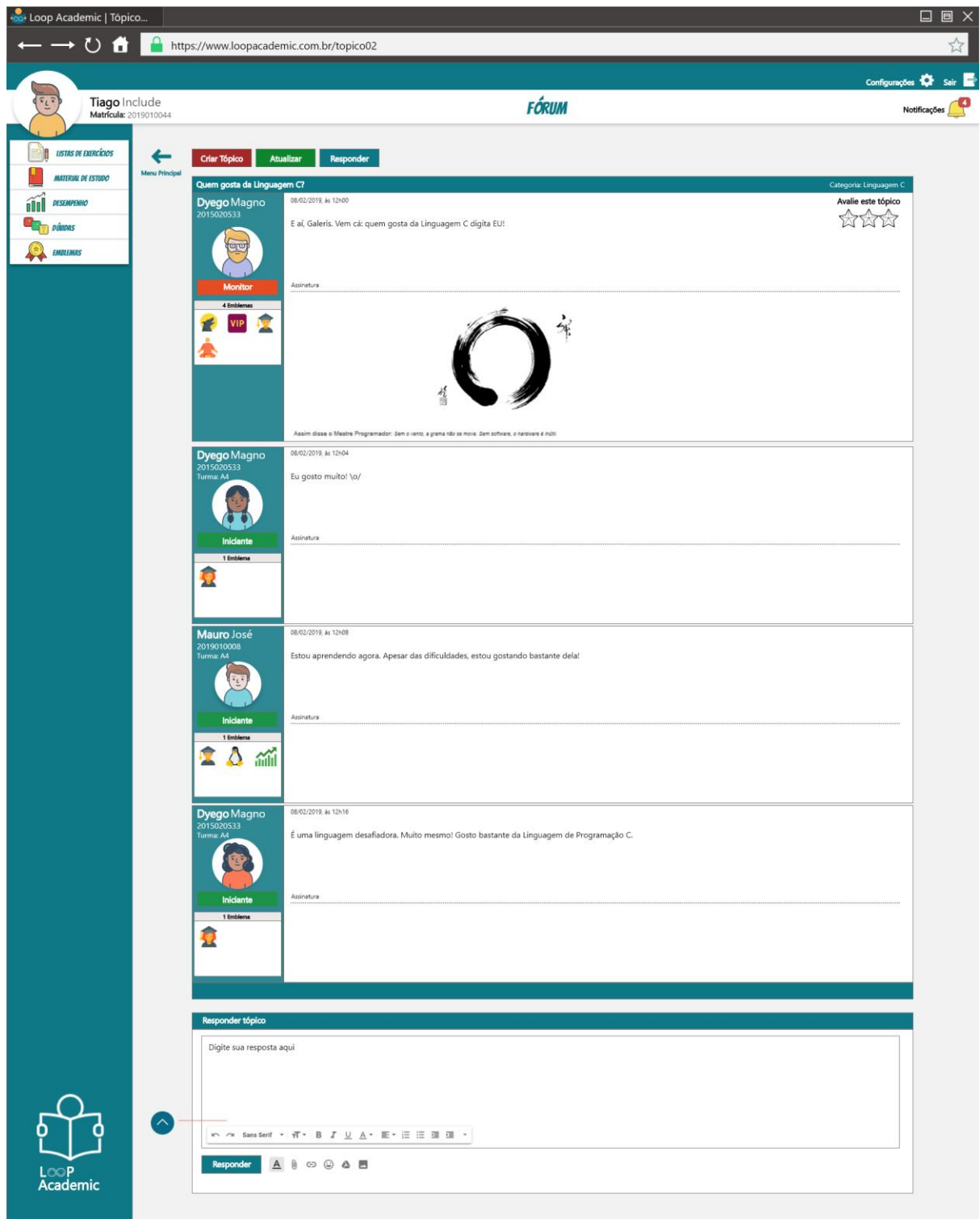
Fonte: Autoria própria, 2019

Figura 38. Protótipo de alta fidelidade: criar tópico no fórum



Fonte: Autoria própria

Figura 39. Protótipo de alta fidelidade: tópico do fórum



Fonte: Autoria própria, 2019

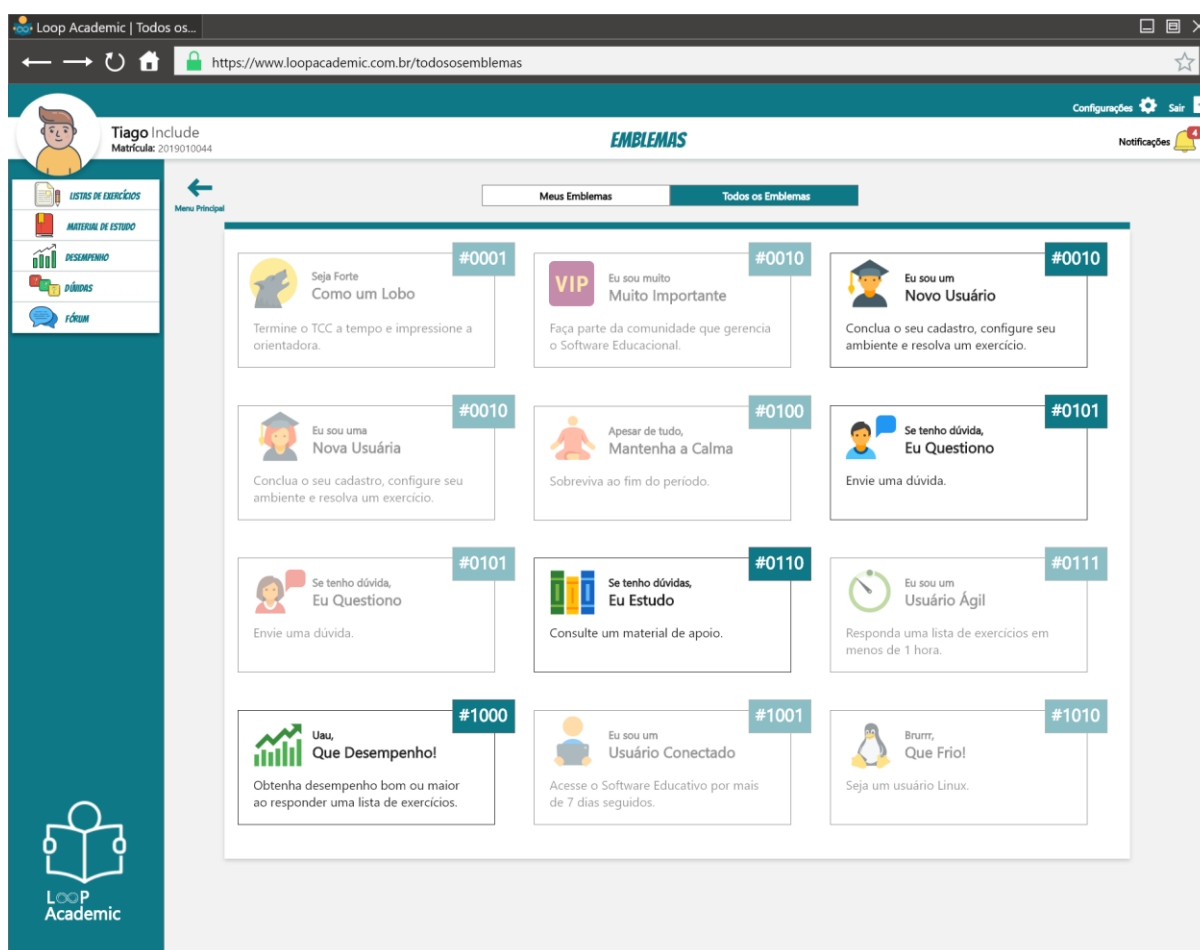
4.4.2.6 Emblemas

Por fim, ao clicar no item “Emblemas”, o aluno é direcionado a uma tela que contém todos os emblemas obtidos e não obtidos. Os Emblemas consistem em pequenos ícones que representam conquistas no *Loop Academic*.

O objetivo dos emblemas consiste em estimular o aluno a realizar determinadas tarefas, baseadas nas funcionalidades do *Loop Academic*, para obtê-las. Um aspecto interessante na obtenção dos emblemas está no fato de que esses podem ser apresentados à todos os usuários no Fórum, o que incentiva ainda mais a vontade de obtê-los.

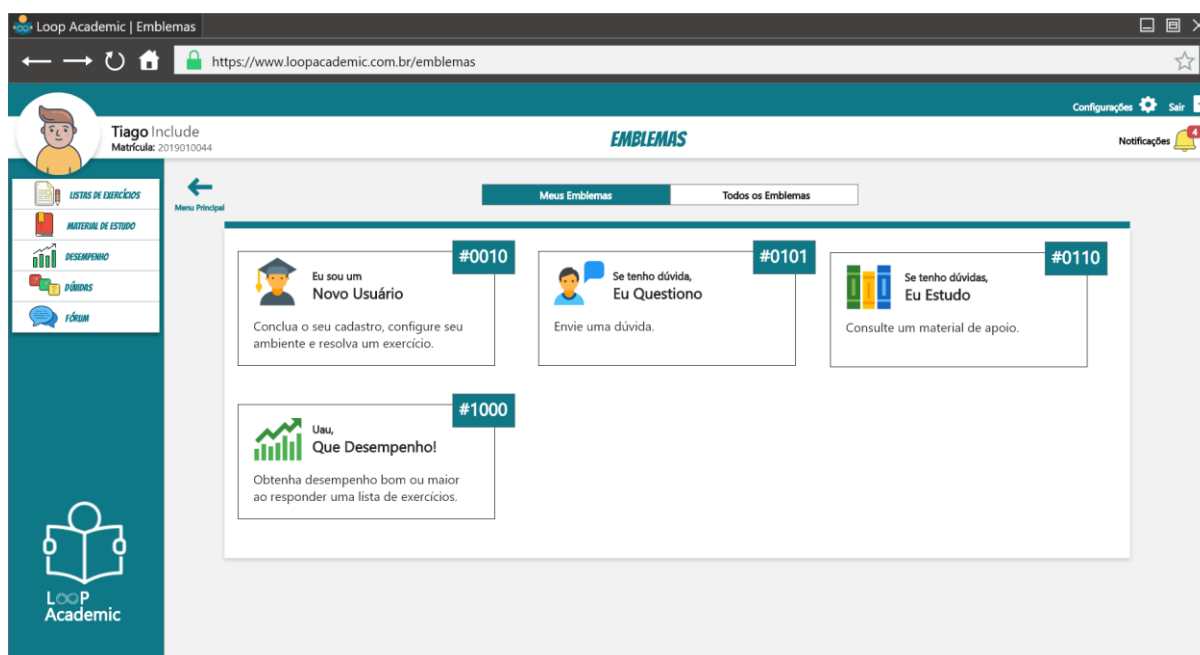
A Figura 40 apresenta todos os emblemas disponíveis, incluindo aqueles obtidos e não obtidos. Já a Figura 41 apresenta todos os emblemas obtidos.

Figura 40. Protótipo de alta fidelidade: emblemas disponíveis



Fonte: Autoria própria

Figura 41. Protótipo de alta fidelidade: Emblemas Obtidos



Fonte: Autoria própria

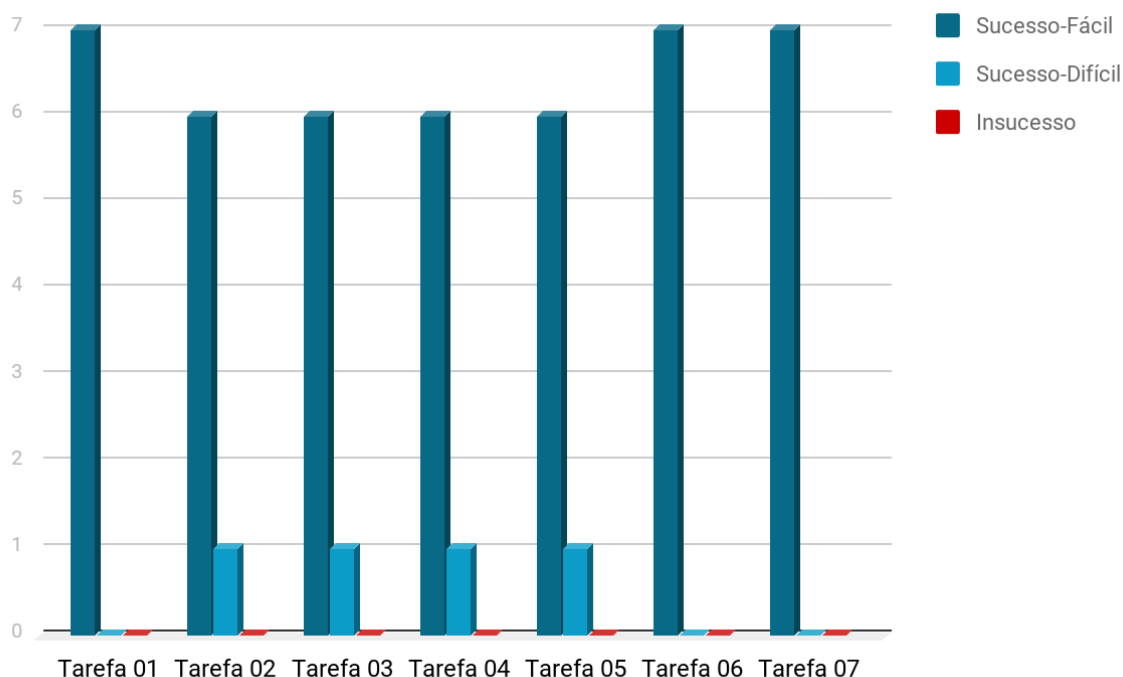
4.5 RESULTADOS DO TESTE DE USABILIDADE

Nas subseções a seguir, são apresentados e discutidos os resultados das três etapas do teste de usabilidade, realizadas com o protótipo de alta fidelidade do *Loop Academic*. As avaliações realizadas obtiveram a participação de 7 (sete) alunos devidamente matriculados na disciplina Algoritmos e Laboratório de Algoritmos do curso de BTI na UFERSA, *Campus Pau dos Ferros*.

4.5.1 Níveis de Acerto

O Gráfico 04 apresenta o resultado dos Níveis de Acerto para as tarefas definidas na Subseção 3.3. Observa-se que nas Tarefas 02, 03, 04 e 05 um aluno (14,3 %) afirmou ter obtido Sucesso-Difícil, enquanto que os outros 6 (seis) alunos (85,7 %) afirmaram terem tido Sucesso-Fácil. Os motivos que levaram à dificuldade do aluno foram fornecidos diretamente no formulário. Entretanto, como houve acompanhamento dos alunos no momento de execução da tarefa, outros defeitos, além dos citados pelo aluno com dificuldade, foram observados e estão definidos no Quadro 04.

Gráfico 04. Níveis de Acerto



Fonte: Autoria própria

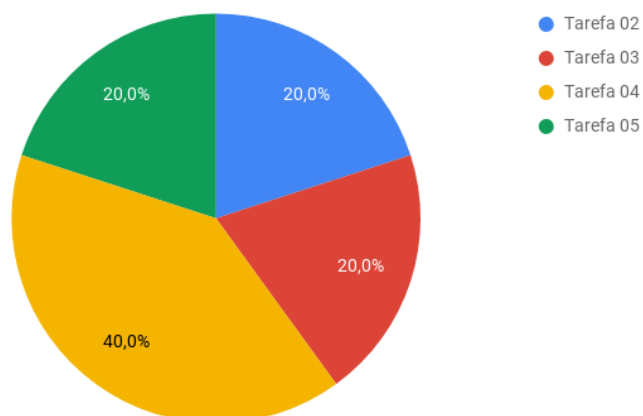
Quadro 04. Defeitos identificados no teste de usabilidade

Nº	Atividade	Descrição do Defeito
01	Atividade 02 - Resolver Exercício	Não ficou evidente para um usuário a diferença entre os botões “Enviar” e “Salvar”
02	Atividade 03 - Consultar Material de Apoio	A navegabilidade entre os itens do material de apoio não ficou clara para um usuário.
03	Atividade 04 - Consultar Desempenho	Um dos usuários alegou ter sentido dificuldades na compreensão dos dados.
04	Atividade 04 - Consultar Desempenho	Todos os usuários tentaram clicar nos exercícios apresentados no quadro, imaginando haver algum tipo de interação.
05	Atividade 05 - Enviar Dúvida	Um dos usuários sentiu dificuldades para localizar o item do menu que permitia-lhe enviar uma dúvida.

Fonte: Autoria própria

Foi calculado também o percentual de defeitos, tido como a razão entre a quantidade de defeitos encontrados em uma determinada tarefa e o total de defeitos encontrados no teste (VALENTIM, 2018). O Gráfico 05 apresenta o percentual de defeitos encontrados por cada tarefa.

Gráfico 05. Percentual de Defeitos.



Fonte: Autoria própria

A partir do Gráfico 05, nota-se que a Tarefa 04 - Consultar Desempenho obteve o maior percentual de erros (40%), enquanto que as demais obtiveram cada qual, 20 % .

4.5.2 Tempo de Execução

O Tempo de Execução foi uma medida avaliativa que visou determinar o tempo de realização para cada tarefa citada na Subseção 3.3. O Quadro 05 apresenta os tempos de execução das tarefas para cada usuário. Ressalta-se que os tempos cronometrados estão no formato minutos:segundos:milésimos.

Quadro 05. Tempos de execução das tarefas

Tarefas	Usuário						
	U01	U02	U03	U04	U05	U06	U07
T01	1:35:36	2:01:14	1:38:32	1:56:24	1:12:29	0:43:04	0:43:00
T02	5:28:06	1:27:31	3:40:23	1:55:35	3:58:19	2:11:50	1:56:53
T03	2:13:07	1:18:26	1:40:07	3:34:58	2:07:14	1:14:13	2:15:27
T04	1:59:01	2:48:46	1:03:24	2:20:11	0:58:18	0:42:10	1:28:49
T05	2:00:32	0:39:29	0:59:18	0:58:20	0:42:34	0:35:23	1:01:02
T06	1:29:30	1:24:36	1:29:56	1:38:44	1:44:26	1:07:03	1:12:29
T09	0:50:35	1:08:46	1:35:32	1:34:38	0:27:32	0:44:36	1:00:20
TOTAL	15:36:27	10:48:48	12:07:12	13:58:50	11:10:52	7:18:19	9:38:00

Fonte: Autoria própria

Pode-se notar que o maior tempo registrado (5:28:06) ocorreu com o usuário U01 ao executar a Tarefa T02 - Responder Exercício. Justifica-se a demora devido ao fato dessa tarefa ser uma das principais realizadas pelo *Loop Academic*, o que, consequentemente, exigiu uma maior exploração dos aspectos presentes na interface.

O segundo e terceiro mais demorados tempos registrados (3:58:19 e 3:40:23, respectivamente) também ocorreram durante a execução da Tarefa T02 com os usuários U05 e U03, respectivamente. Os tempos demandados para realizar a tarefa pelos dois usuários reforçam o que fora dito na avaliação da Tarefa T02.

Pode-se notar que o menor tempo registrado (0:27:32) ocorreu com o usuário U05 durante a realização da Tarefa T09 - Consultar Emblema. Justifica-se o ocorrido devido ao pequeno escopo dessa tarefa, que não demandava muitos esforços para ser executado.

O segundo e terceiro menores tempos registrados (0:35:23 e 0:42:10, respectivamente) ocorreram durante a execução das Tarefas T05 - Enviar Dúvida e T04 - Consultar Desempenho.

No tocante aos tempos de execução para cada tarefa, temos que o intervalo de tempo entre o maior e o menor tempos é um dado que reflete possíveis dificuldades na interface. O Quadro 06 apresenta os intervalos de tempo para cada tarefa realizada.

Quadro 06. Intervalos de tempo para cada tarefa

Tarefa	Execução		Intervalo de tempo (próximo)
	Maior Tempo	Menor Tempo	
T01	2:01:14	0:43:00	$\cong$ 1:18
T02	5:28:06	1:27:31	$\cong$ 4:01
T03	3:34:58	1:14:13	$\cong$ 2:21
T04	2:48:46	0:42:10	$\cong$ 2:06
T05	2:00:32	0:35:23	$\cong$ 1:25
T06	1:44:26	1:07:03	$\cong$ 0:37
T07	1:35:32	0:27:32	$\cong$ 1:08

Fonte: Autoria própria

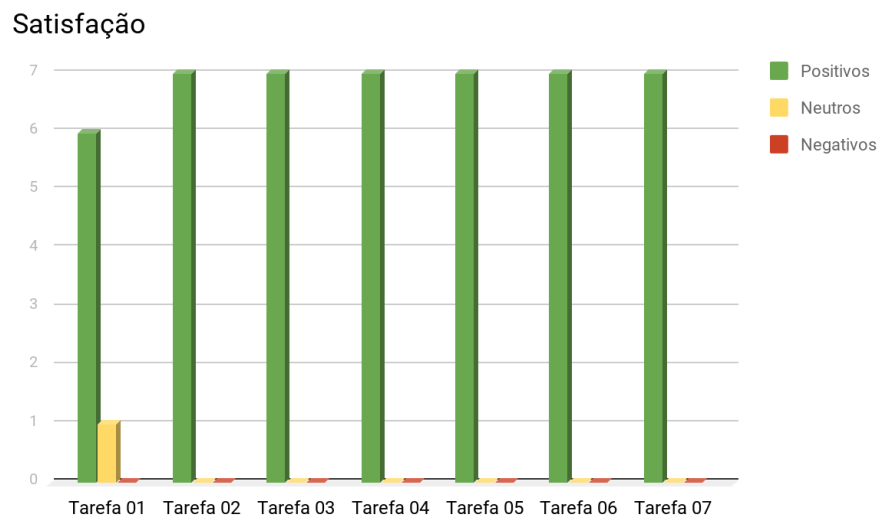
Diante dos dados apresentados no Quadro 05, destaca-se o intervalo de tempo para a Tarefa T02, que apresentou um elevado intervalo de tempo (aprox. 4:01). Esse fato indica necessidade de adequação quanto aos aspectos presentes na interface da Tarefa 02, tendo em

vista que um usuário demandou bastante tempo executá-la. As Tarefas T03 e T04 também carecem de atenção devido ao fato de terem apresentado razoável intervalo de tempo (aprox. 2:21 e 2:06, respectivamente).

4.5.3 Qualidade Afetiva

Para a análise dos resultados da qualidade afetiva da interface, foi adotada a estratégia proposta por Valentim (2018), que consiste na contagem dos votos para as telas avaliadas em cada tarefa e posterior classificação destes através dos votos obtidos pela Escala SAM e classificados em: (i) positivos, votos marcados à direita da coluna central de cada uma das três dimensões (Satisfação, Controle e Motivação); (ii) neutros, votos marcados na coluna central de cada uma das dimensões; (iii) negativos, votos marcados à esquerda da coluna central de cada uma das dimensões. Em seguida, somá-los e contabilizá-los em um resultado abrangente do escopo analisado. Diante disto, os Gráficos 06, 07 e 08 apresentam os resultados para as três dimensões avaliadas pela Escala SAM: Satisfação, Controle e Motivação.

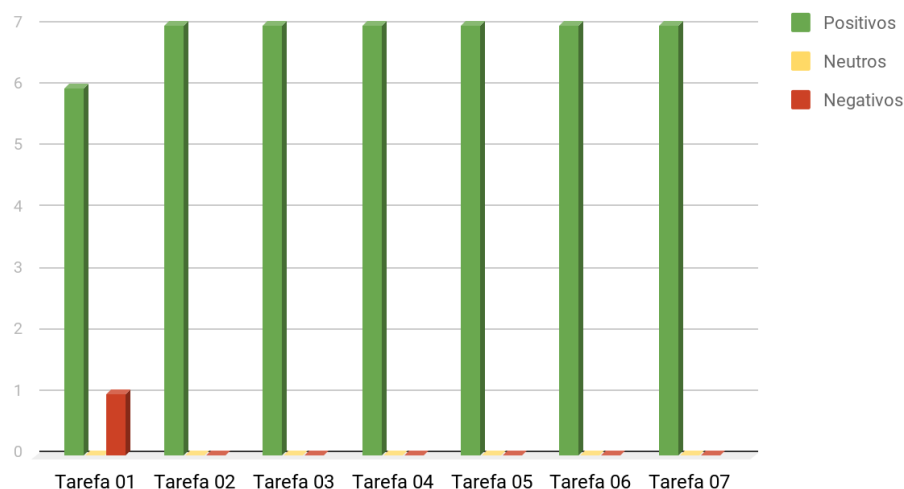
Gráfico 06. Resultados da Escala SAM: Satisfação



Fonte: Autoria própria

Gráfico 07. Resultados da Escala SAM: Controle

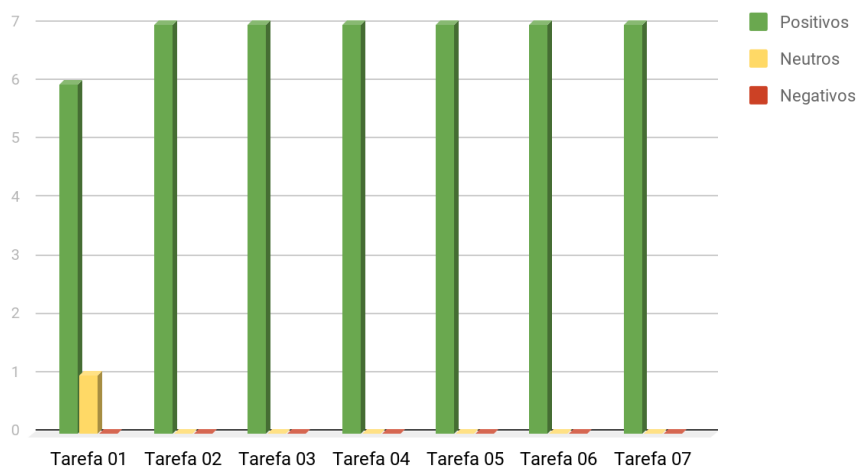
Controle



Fonte: Autoria própria

Gráfico 08. Resultados da Escala SAM: Motivação

Motivação



Fonte: Autoria própria

Quanto à Satisfação, Sentimento de Controle e Motivação dos Usuários em relação ao protótipo de alta fidelidade, os resultados obtidos, conforme apontam os gráficos acima (ver Gráficos 06, 07 e 08), foram positivos para quase todas as telas - com exceção das telas da Tarefa 01, que apresentaram duas avaliações neutras para as dimensões Satisfação e Motivação, e uma avaliação negativa para a dimensão Controle.

O grande quantitativo de avaliações positivas pode ser justificado pelas opiniões feitas pelos alunos, que são apresentadas no Quadro 07.

Quadro 07. Opiniões dos participantes do teste de usabilidade

Nº	Opinião realizada
01	“Gostei da interface, achei nela apenas aquilo que um aluno realmente necessita”
02	“A tela inicial da plataforma mostra uma tela limpa com boa organização dos ícones, bem centralizado, trazendo tudo de uma forma simples”
03	“Na tela de resolução de exercícios, a questão ta bem explicativa, tenho acesso à dicas, código de apoio e outras coisas”
04	“Gostei, achei que vai facilitar bastante na resolução de listas”
05	“Gostei das opções de escolha de vídeo-aulas, etc. Me motivaram bastante porque é o conteúdo certo para minhas dúvidas”
06	“Organizado para os estudos”
07	“Consegui acessar tudo com facilidade”
08	“A interface apresenta bom entendimento, pois mostra de forma centralizada e simples cada função ali proposta, e os menus são bastante intuitivos”
09	“Fiquei motivado com os resultados” (refere-se ao desempenho)”
10	“O meu nível de desempenho me motiva a estudar. Quando ta baixo, tenho que estudar. Quando ta alto, tenho que continuar estudando para manter”
11	“Achei que iria me ajudar bastante, pois além de mostrar como foi o meu desempenho, me ajuda mostrando o material necessário para com que eu consiga melhorar o meu desempenho”
12	“Interessante a forma de usar emblemas, causa maior motivação em continuar na plataforma, e continuar com o estudo, as telas apresentam de forma direta onde tem que ir, e é limpa e bem organizada”
13	“Gostei do fato de [o envio de dúvidas] ser anônimo, pois isso me deixa mais segura a tirar uma dúvida, a interface é de fácil acesso”
14	“Mostra de forma simples como enviar dúvidas para o professor, também outro aspecto interessante é que tem a opção de anexar arquivos onde pode enviar a questão que está com dificuldade”
15	“Apresenta de forma simples como criar um tópico e como poder inserir dúvidas para que outras pessoas respondam, também é legal porque mostra os emblemas ganhos na plataforma mostrando que quem responde tem maior relevância na questão”
16	“Me senti motivado a conquistar mais emblemas”
17	“Gostei bastante de ver minhas conquistas e desbloquear conquistas. Me deu vontade de desbloquear elas”

18	“É motivador, pois vai nos levar a responder mais questões para ganhar esses títulos” [refere-se à conquista de emblemas]
19	“Tela bem projetada, mostrando todos os emblemas conquistados e os outros que faltam, bem organizados e intuitivos”
20	“Os emblemas podem acabar aumentando os interesses dos usuários, pois é como se fosse uma recompensa dentro do ‘software’”

Fonte: Autoria própria

As avaliações neutras para as telas que compuseram a Tarefa 01 - Realizar Login foram reflexo da opinião de um aluno, que justificou a avaliação neutra para a dimensão Satisfação devido à quantidade de texto na tela de boas-vindas e a avaliação neutra para a dimensão Motivação devido à formalidade da tela de cadastro de um aluno, o que não o deixou motivado a concluir o cadastro. O mesmo aluno também justificou a sua avaliação negativa para a dimensão Controle, cuja nota foi atribuída à tela de cadastro de um aluno (ver Figura 20), como decorrente da ausência de um mecanismo facilitador do cadastro, como o Gmail ou até mesmo Facebook. Essa ausência não lhe deu autonomia para definir a forma como queria ser cadastrado, resultando numa sensação de controle.

5. CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS

Este trabalho propôs o desenvolvimento do protótipo de um Software Educacional, denominado *Loop Academic*, direcionado ao auxílio no processo de ensino-aprendizagem de programação introdutória em cursos de nível superior, em virtude dos problemas percebidos no processo de ensino-aprendizagem de programação introdutória em cursos de nível superior no Brasil, e de modo mais específico no curso de Bacharelado em Tecnologia da Informação da Universidade Federal Rural do Semi-Árido, campus Pau dos Ferros - RN. Assim, foi realizado um *survey* com alunos, professores e monitores da disciplina Algoritmos do referido curso da UFRSA, *campus* Pau dos Ferros. Em seguida, os requisitos foram definidos e os *personas* elaborados, para representar os perfis dos potenciais usuários do *Loop Academic*. Logo após, foram desenvolvidos os protótipos de baixa e de alta fidelidade, e aplicados Testes de Usabilidade para validação dos protótipos.

Os resultados obtidos para cada etapa da pesquisa, revelaram que: (i) os dados obtidos com o *survey* permitiram avaliar o escopo do trabalho, fornecendo informações úteis para análise da viabilidade do *Loop Academic* e a elicitação de seus requisitos; (ii) os requisitos (funcionais e de usabilidade) adequam-se à realidade interna e externa dos contextos estudantis relacionados ao processo de ensino-aprendizagem de programação introdutória, tendo em vista sua similaridade com trabalhos relacionados em vigor; (iii) os *personas* contribuíram para o desenvolvimento do protótipo, pois estabeleceram duas visões diferentes- visão de um aluno novato e suas dificuldades, e visão de um aluno não-aprovado e suas dificuldades; (iv) os protótipos foram desenvolvidos levando em consideração os requisitos definidos, *personas* e o maior alcance possível da usabilidade, bem como a elaboração dos protótipos de baixa fidelidade foi complementar à elaboração dos protótipos de alta fidelidade; (v) a escolha pelo teste de usabilidade como forma de avaliar a usabilidade do protótipo do *Loop Academic* permitiu avaliar aspectos relacionados a usabilidade do protótipo, com resultados que apontaram avaliações em seu pico máximo, como foi o caso da escala SAM, que obteve quase todas as máximas avaliações para as três dimensões (Satisfação, Controle e Motivação).

Considerando todo o exposto, acredita-se que este trabalho tenha alcançado seus objetivos. Entretanto, há ainda muito a ser explorado, o que fomenta a elaboração de trabalhos futuros, a citar: (i) aplicação da Inspeção de Usabilidade nos protótipos desenvolvidos, para validação das funcionalidades propostas; (ii) prototipação dos módulos Professor e Monitor, a partir do *survey* aqui apresentado, tendo em vista que foi não possível desenvolvê-los nesta pesquisa;-(iii) desenvolver e implementar os módulos Aluno, Professor e Monitor, tendo em

vista a dependência dos três módulos para o funcionamento do *Loop Academic*; (iv) realizar testes nos módulos implementados, o que permitirá a análise dos aspectos relacionados a qualidade pedagógica e funcional do *Loop Academic*; (v) aplicação de estudos de casos com o *Loop Academic*.

REFERÊNCIAS

- ABREU, F.; ALMEIDA, A.; BARREIROS, E.; SARAIVAL, J.; SOARES, S.; ARAUJO, A.; HENRIQUE, G. Métodos, Técnicas e Ferramentas para o Desenvolvimento de Software Educacional: um Mapeamento Sistemático. **Anais do XXIII Simpósio Brasileiro de Informática na Educação (SBIE 2012)**, Rio de Janeiro - RJ, p.1-10, 26-30 nov. 2012. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).
- ALMEIDA, A.; GOMES, T. S.; LEAL, V. de Q.; GOMES, R. C. de M.; GOMES, L. de Q. L. Indicadores para Avaliação de Software Educacional com base no guia GDSM (Goal Driven Software Measurement). **Anais do XXIX Simpósio Brasileiro de Informática na Educação (SBIE 2018)**, [s.l.], p.21-30, 28 out. 2018. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).
- ARNOLD, T. C. **Além da Interação Homem-computador: O Design de Interação**, seus Processos e Metas, em Busca da Satisfação do Usuário Final. 2011. Disponível em: <http://tatianearnold.freetzi.com/design_de_interacao.pdf>. Acesso em: 23 fev. 2019.
- BARBOSA, A. de A.; FERREIRA, D. Í. S.; COSTA, E. B. Influência da Linguagem no Ensino Introdutório de Programação. **Anais do XXV Simpósio Brasileiro de Informática na Educação (SBIE 2014)**, [s.l.], p.612-621, 2014. Sociedade Brasileira de Computação - SBC.
- BARBOSA, D. F. **Uma Estratégia de Apoio à Institucionalização da Usabilidade em Ambientes de Desenvolvimento Ágil**. 2008. 180 f. Dissertação (Mestrado) - Curso de Informática Aplicada, Universidade de Fortaleza - UNIFOR, Fortaleza, 2008.
- BARBOSA, S. D. J.; SILVA, B. S. da. (2010). **Interação Humano-Computador**. 1. ed. Rio de Janeiro: Elsevier, 2010, 408 p.
- BASSANI, P. S.; PASSERINO, L. M.; PASQUALOTI, P. R.; RITZEL, M. I. Em busca de uma proposta metodológica para o desenvolvimento de software educativo colaborativo. **Novas Tecnologias na Educação**, [s.l.], v. 4, n. 1, p.1-10, jul. 2006.
- BENYON, D.. **Interação Humano-Computador**. 2. ed. [s.i]: Pearson, 2011.
- BOURQUE, P.; FARLEY, R. E. D. **SWEBOK: Guide to the Software Engineering Body of Knowledge**. 3. ed. [s.i]: Ieee, 2014. (IEEE - Computer Society). Disponível em: <<http://beamphys.triumf.ca/info/SWEBOKv3.pdf>>. Acesso em: 25 jan. 2019.
- BRITO JÚNIOR, A.; MADEIRA, C. XP & Skills: gamificando o processo de ensino de introdução a programação. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015)**, [s.l.], p.1124-1133, 26 out. 2015. Sociedade Brasileira de Computação - SBC.
- BRITO JÚNIOR, O.; AGUIAR, Y.; TAVARES, T. Abordagens para Avaliação de Softwares Educativos e sua Coerência com os Modelos de Qualidade de Software. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)**, [s.l.], p.270-279, 7 nov. 2016. Sociedade Brasileira de Computação - SBC.

CAFÉ, M.; COSTA, F. A de M.; SILVA, V. L.; TERRA, D. C. Lord of Code: Uma Ferramenta de Apoio ao Ensino da Programação. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)**, [s.l.], p.1316-1320, 7 nov. 2016. Sociedade Brasileira de Computação - SBC.

CAMARGO, S. da. S.; CECHINEL, C. APRENDE - Um Novo Sistema Tutor Inteligente para Auxiliar nas Dificuldades dos Conteúdos de Algoritmos e Programação. **Anais do XII Workshop Sobre Educação em Computação (WEI 2014)**, [s.l.], p.1416-1425, 2014.

CAMPOS, R. L. B. L. “Metodologia ERM2C: para melhoria do processo de ensino-aprendizagem de lógica de programação”. Workshop de Ensino em Informática – WEI. Belo Horizonte – MG. 2010.

CAMPOS, A.; GARDIMAN, R. Q.; MADEIRA, C. Uma Ferramenta Gamificada de Apoio à Disciplina Introdutória de Programação. **Anais do XIII Workshop Sobre Educação em Computação (WEI 2015)**, Recife, PE, p.1-10, 2015

CETIC.BR (Brasil). Núcleo de Informação e Coordenação do Ponto Br. **TIC EDUCAÇÃO 2017 - Pesquisa Sobre o Uso das Tecnologias de Informação e Comunicação nas Escolas Brasileiras**. São Paulo, SP: Comitê Gestor da Internet no Brasil, 2018. 436 p. Disponível em: https://www.cetic.br/media/docs/publicacoes/2/tic_edu_2017_livro_eletronico.pdf. Acesso em: 08 dez. 2018.

COELHO, O. B.; SANTORO, D. M. A Sinergia entre as Heurísticas de Usabilidade de Software e as Heurísticas de Ensino-Aprendizagem do ponto de vista da Educação à Distância Mediada pela Web. **Anais do XIII Simpósio Brasileiro de Informática na Educação (SBIE 2002)**, [s.l.], p.273-281, 2002. Sociedade Brasileira de Computação - SBC.

COOMANS, S.; LACERDA, G. S. PETESE, a Pedagogical Ergonomic Tool for Educational Software Evaluation. **Procedia Manufacturing**, [s.l.], v. 3, p.5881-5888, 2015. Elsevier BV.

COSTA, E. C. da. A Importância da Engenharia de Requisitos no Processo de Desenvolvimento de Sistemas de Informação. **Revista Interface Tecnológica**, [s.l.], v. 15, n. 1, p.203-214, 12 jul. 2018. Interface Tecnológica.

COUTINHO, C.; LISBOA, E.. Sociedade da Informação, do Conhecimento e da Aprendizagem: Desafios para Educação no Século XXI. **Revista de Educação**, [s.l.], v. XVIII, n. 1, p. 5-22, 2011.

ENGHOLM JÚNIOR, H. **Engenharia de Software na prática**. São Paulo: Novatec, 2010.

FANTIN, Kátia. **Metodologia de Avaliação de Software Educacional**. 2017. 21 f. TCC (Graduação) - Curso de Sistemas de Informação, Universidade de Caxias do Sul - UCS, Caxias do Sul, RG, 2017.

GIRAFFA, L. M .M.; MORA, M. C. Evasão na Disciplina de Algoritmo e Programação: Um Estudo a Partir dos Fatores Intervenientes na Perspectiva do Aluno. In: **Tercera Conferencia sobre el Abandono en la Educación Superior** (III CLABES, 2013, Mexico. Tercera

Conferencia sobre el Abandono en la Educación Superior (III CLABES. Madrid: EUIT de Telecomunicación, 2013. v. 1. p. 188-197.

GUO, F. Y.; SHAMDASANI, S.; RANDALL, B. “Creating Effective Personas for Product Design: Insights from a Case Study”. In: **International Conference in Internationalization, Design and Global Development**. Springer Berlin Heidelberg, p. 37-46, 2011.

HOLANDA, W. D. de.; COUTINHO, J. C. da S.; FONTES, L. M. de O. Uma Intervenção Metodológica para Auxiliar a Aprendizagem de Programação Introdutória: um estudo experimental. **Anais dos Workshops do VII Congresso Brasileiro de Informática na Educação (CBIE 2018)**, [s.l.], p.699-708, 28 out. 2018. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).

JESUS, A. de; BRITO, G. S. Concepção de Ensino-Aprendizagem de Algoritmos e Programação de Computadores: A Prática Docente. In: **ENINED - Encontro Nacional de Informática e Educação**, 1., 2009. Cascavel, PR: Universidade Estadual do Oeste do Paraná – UNIOESTE, 2009. p.132-141.

LANG, P. Behavioral treatment and bio-behavioral assessment: computer applications. In: **J. B. Sidowski, J. H. Johnson; T. A. Williams (Eds.)**, Technology in mental health care delivery systems. Norwood, NJ: Ablex, 1980.

LIMA, J. F. S. de.; LEAL, V. de Q.; GOMES, R. C. de M.; GOMES, L. de Q. L. Quali-EDU: Um processo de avaliação da qualidade de software educacional. **Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)**, [s.l.], p.229-238, 26 out. 2015. Sociedade Brasileira de Computação - SBC.

MATERA, M.; COSTABILE, M. F.; GARZOTTO, F.; PAOLINI, P. SUE inspection: an effective method for systematic usability evaluation of hypermedia. **IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans**, [s.l.], v. 32, n. 1, p.93-103, 2002. Institute of Electrical and Electronics Engineers (IEEE).

MEDEIROS, M. F.; VALLETTA, D.; MAGAGNIN, E. B.; RIBEIRO, E. M. P.; DABOIT, K. L. dos S. A Atenção Voluntária na Construção de Conceitos Trigonométricos em Ambientes de Geometria Dinâmica. **Revista Brasileira de Informática na Educação**, [s.l.], v. 25, n. 01, p.77-93, 30 abr. 2017. Comissão Especial de Informática na Educação.

MORAES, R. S.; WEBBER, C. G. Uso das Tecnologias da Informação na Motivação dos Alunos para as Aulas de Química. **Scientia Cum Industria**, [s.l.], v. 5, n. 2, p.95-102, 27 set. 2017. Universidade Caxias do Sul.

MOURA, V. A. B. de.; PERES, L. M. Avaliação do impacto da retroação na aprendizagem apoiada por uma ferramenta educacional. **Revista Brasileira de Informática na Educação**, [s.l.], v. 25, n. 01, p.60-76, 30 abr. 2017. Comissão Especial de Informática na Educação.

NASCIMENTO, I.; GASPAR, W.; CONTE, T.; GADELHA, B.; OLIVEIRA, E. H. T. Melhor prevenir do que remediar: Avaliando usabilidade e UX de software antes de levá-lo para a sala de aula. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)**, [s.l.], p.806-815, 7 nov. 2016. Sociedade Brasileira de Computação - SBC.

- NIELSEN, J.. Heuristic evaluation. **Usability inspection methods**, v. 17, n. 1, p.25-62, 1994.
- PAES, R. de B.; MALAQUIAS, R.; GUIMARÃES, M.; ALMEIDA, H. Ferramenta para a Avaliação de Aprendizado de Alunos em Programação de Computadores. **II Congresso Brasileiro de Informática na Educação (CBIE 2013)**, [s.l.], p.203-212, 25 nov. 2013. Sociedade Brasileira de Computação - SBC.
- PAULA, A. C. de.; VERGARA, L.; LUZ, R. M. da; VIALI, L. Softwares Educacionais para o Ensino de Física, Química e Biologia. **Revista Ciências&Ideias**, [s.l.], v. 5, n. 1, p.107-121, abr. 2014.
- PEREIRA, W. S.; DA SILVA, R. S. T.; CARDOSO FILHO, R. J.; SILVA, W. R. de A.; AGUIAR, Y. P. C.; DANTAS, V. F. Avaliação de Software Educativo: Análise de Abordagens para Definição de Diretrizes. In: **XXI Conferência Internacional Sobre Informática na Educação (TISE 2016)**. Santiago de Chile: Nuevas Ideas En Informática Educativa, 2016. v. 12, p. 557 – 562.
- PEREIRA JÚNIOR, J. C. R.; RAPKIEWICZ, C. E.; DELGADO, C.; XEXEO, J. A. M. Ensino de algoritmos e programação: uma experiência no nível médio. In: **Anais do XIII Workshop de Ensino em Informática (WEI 2005)**, São Leopoldo, 2005, p. 2352-2362.
- POSADA, J. E. G.; BUCHDID, S. B.; BARANAUSKAS, M. C. C. A informática na educação: o que revelam os trabalhos publicados no Brasil. **Revista Brasileira de Informática na Educação**, [s.l.], v. 24, n. 1, p.142-155, 18 ago. 2016. Comissão Especial de Informática na Educação.
- PRESSMAN, R. S. **Engenharia de Software: Uma Abordagem Profissional**. 8. ed.: Mcgraw-Hill, 2016. 968 p.
- PRUITT, J.; GRUDIN, J. Personas: Practice and Theory. **Proceedings Of The 2003 Conference On Designing For User Experiences - Dux '03**, [s.l.], p.1-8, 2003. ACM Press.
- QUEIROZ, J. V.; RODRIGUES, L. M.; COUTINHO, J. C. Um relato dos Fatores Motivacionais na Aprendizagem de Programação na Perspectiva de Alunos iniciantes em Programação da Universidade Federal Rural do Semi-Árido *Campus* Pau dos Ferros-RN. In: **III ENCONTRO DE COMPUTAÇÃO DO OESTE POTIGUAR - ECOP**, 2018, Pau dos Ferros - RN. v. 2, p. 162 – 168.
- RAPOSO, E. H. S.; DANTAS, V. O Desafio da Serpente - Usando gamification para motivar alunos em uma disciplina introdutória de programação. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)**, [s.l.], v. 5, n. 27, p.577-586, 7 nov. 2016. Sociedade Brasileira de Computação - SBC.
- ROCHA, E. M. **Tecnologias Digitais e Ensino de Matemática: Compreender para Realizar**. 2008. 200 f. Tese (Doutorado) - Curso de Educação, Universidade Federal do Ceará - UFC, Faculdade de Educação, Programa de Pós-Graduação em Educação Brasileira, Fortaleza - CE, 2008.
- ROCHA, H.; BARANAUSKAS, M. **Design e Avaliação de Interfaces Humano-Computador**. NIED, UNICAMP, Campinas, SP. 2003, 244 p.

ROGERS, Y.; SHARP, H.; PREECE, J.. **Design de Interação: Além da Interação Humano-Computador**. 3. ed. [s.i]: Bookman, 2013.

SANTANCHÈ, A.; TEIXEIRA, C. A. C. Mais pontes e menos ilhas: estratégias para a integração de software educacional. **Anais do XIII Simpósio Brasileiro de Informática na Educação (SBIE 2002)**, [s.l.], p.291-301, nov. 2002. Sociedade Brasileira de Computação - SBC.

SELIVON, M.; BEZ, J. L.; TONIN, N. A. *URI Online Judge Academic: Integração e Consolidação da Ferramenta no Processo de Ensino/Aprendizagem*. **Anais do XIII Sobre Educação em Computação (WEI 2015)**, [s.l.], 2015. CSBC, 2015.

DA SILVA, A. C. B.; DE FRANÇA, R. S. Avaliação de software educativo na formação docente: articulando teoria e prática. **Anais do XX Workshop de Informática na Escola (WIE 2014)**, [s.l.], p.106-115, 3 nov. 2014. Sociedade Brasileira de Computação - SBC.

SILVA, A. C. B. da. **Softwares Educativos: Critérios de Avaliação a partir dos Discursos da Interface, da Esfera Comunicativa e do Objeto de Ensino**. Recife, 2012. 346 f.: Tese (doutorado) - UFPE, Centro de Educação, Programa de Pós-graduação em Educação, 2012.

SILVA, A. R.; VALERIO, M. M. C.; NETO, P. A. M.; CAMPOS FILHO, A. S. Anatomia Digital: Um ambiente virtual de apoio ao processo ensino-aprendizagem. **Anais do XXVIII Simpósio Brasileiro de Informática na Educação (SBIE 2017)**, [s.l.], p.745-755, 27 out. 2017. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC).

SILVA, P.; CAVALCANTE, M. T. C.; FECHINE, J.; COSTA, E. Um Mapeamento Sistemático sobre Iniciativas Brasileiras em Ambientes de Ensino de Programação. **Anais do XXVI Simpósio Brasileiro de Informática na Educação (SBIE 2015)**, [s.l.], p.367-375, 26 out. 2015. Sociedade Brasileira de Computação - SBC.

SILVA, R. S. T.; SILVA, W. R. de A.; CARDOSO FILHO, R. J.; PEREIRA, W. S.; AGUIAR, Y. P. C.; DANTAS, V. F. Aplicação Comparativa de Diferentes Abordagens de Avaliação para o Software Educativo Duolingo: a Complexidade de Escolher uma Abordagem Adequada. **Tecnologias na Educação**, [s.l.], v. 16, n. 8, p.1-17, set. 2016. Semestral.

SOMMERVILLE, I. **Engenharia de software**. 9.ed. São Paulo: Pearson, 2011. xiii, 529p.

TOLEDO, B. de S. **O Uso de Softwares como Ferramenta de Ensino-Aprendizagem na Educação do Ensino Médio/Técnico no Instituto Federal de Minas Gerais**. 2015. 114 f. Dissertação (Mestrado) - Curso de Sistemas de Informação e Gestão do Conhecimento, Universidade FUMEC. Belo Horizonte, BH, 2015.

VALENTIM, N. M. C. **Antecipando a Usabilidade nas Fases Iniciais do Processo de Desenvolvimento de Software**. 2017. 249 f. Tese (Doutorado) - Curso de Computação, Instituto de Computação, Universidade Federal do Amazonas - UFAM, Manaus, AM, 2017.

VALLETTA, D. Educação Digital Brasileira: Possibilidades e Desafios. In: **Reunião Científica Regional da ANPED**, 21., 2016, Curitiba, Pr: ANPED SUL, 2016. p. 1 - 12.

VERMEEREN, A. P. O. S.; ROTO, V.; et al. User experience evaluation methods. **Proceedings Of The 6th Nordic Conference On Human-computer Interaction Extending Boundaries - Nordichi '10**, [s.l.], p.521-530, 2010. ACM Press.

VIEIRA, F. M. S. **Classificação de Softwares Educacionais**. 2007. 3 f. Curso de Mídias na Educação, Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul - UFRGS, Porto Alegre, RG, 2007.

ANEXO A



UNIVERSIDADE FEDERAL RURAL DO SEMIÁRIDO
(UFERSA) - CAMPUS PAU DOS FERROS
CURSO: BACHARELADO EM TECNOLOGIA DA
INFORMAÇÃO

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Você está sendo convidado(a) como voluntário(a) a participar da pesquisa intitulada **“Avaliação de Usabilidade – Erros por Atividade e Escala SAM”**, que compõe o Trabalho de Conclusão de Curso (TCC) intitulado **“Desenvolvimento e Avaliação de um Protótipo de Software Educacional para o Auxílio no Processo de Ensino-Aprendizagem de Programação Introdutória”**, do pesquisador e aluno Dyego Magno Oliveira Souza e sua orientadora Jarbele Cássia da Silva Coutinho. Após receber os esclarecimentos e as informações a seguir, no caso de aceitar fazer parte do estudo, este documento deverá ser assinado em duas vias, sendo a primeira de guarda e confidencialidade do Pesquisador (a) responsável e a segunda ficará sob sua responsabilidade para quaisquer fins.

Em caso de recusa, você não será penalizado (a) de forma alguma. Em caso de dúvida sobre a pesquisa, você poderá entrar em contato com o (a) pesquisador (a) responsável **Dyego Magno Oliveira Souza** através do telefone **(84) 98161-6295** ou através do e-mail **dyegowolf@gmail.com**.

A presente pesquisa é motivada pela **avaliação de usabilidade dos protótipos da plataforma Loop Academic, desenvolvido como TCC do presente pesquisador para obtenção do título de Bacharela em Tecnologia da Informação**. Para a coleta de dados serão utilizados cenários fictícios passíveis de compreensão e execução no protótipo demonstrado pelo pesquisador.

Você será esclarecido(a) sobre a pesquisa em qualquer tempo e aspecto que desejar, através dos meios citados anteriormente. Você é livre para recusar-se a participar, retirar seu consentimento ou interromper a participação a qualquer momento, sendo sua participação voluntária e a recusa em participar não irá acarretar qualquer penalidade.

O(s) pesquisador(es) irá(ão) tratar a sua identidade com padrões profissionais de sigilo e todos os dados coletados servirão apenas para fins de pesquisa. Seu nome ou o material que indique a sua participação não será liberado sem a sua permissão. Você não será identificado(a) em nenhuma publicação que possa resultar deste estudo.

Ciente e de acordo com o que foi anteriormente exposto, eu _____, estou de acordo em participar da pesquisa intitulada “Avaliação de Usabilidade – Erros por Atividade e Escala SAM”, de forma livre e espontânea, podendo retirar meu consentimento a qualquer momento.

_____, de _____ de 20____

Assinatura do responsável pela pesquisa

Assinatura do participante