



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Feedbacks formativos e motivacionais em uma disciplina de introdução à computação

Lucas Raphael Ferreira de Miranda

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.a Dr.a Maristela Terto de Holanda

Brasília
2023

Dedicatória

Dedico este trabalho a minha família e meus amigos, presentes em todos os momentos dessa longa jornada. À família pelo massivo suporte diário. E aos amigos, minha segunda família, por estarem sempre presentes, pelos abraços, motivações e descontrações. Nada seria possível sem tal rede de apoio, pilares de minha vida.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

A alta taxa de reprovação das disciplinas de ensino introdutório de programação é um desafio presente no Brasil e em várias universidades do mundo, incluindo a Universidade de Brasília com a disciplina Introdução à Ciência da Computação. Neste contexto, esse artigo possui como objetivo principal, a implementação de feedbacks elaborados e personalizados para os estudantes ao realizarem as listas de exercícios que a ementa da disciplina prevê, apresentando, assim, um progresso no desenvolvimento do *plugin* Assistente de Programação Introdutória (AsPIn). Os feedbacks foram validados com alunos da Universidade de Brasília no segundo semestre de 2022. Na pesquisa final, 87,5% do grupo concordou com a eficácia do feedback elaborado em seu desempenho na disciplina.

Palavras-chave: Feedback, Plugin, Moodle, Ensino de programação, Aprendizado de programação, Introdução à Ciência da Computação.

Abstract

The high failure rate of introductory programming courses is a challenge present in Brazil and in several universities around the world including the University of Brasília with the Introduction to Computer Science course. In this context, this article has as its main objective the implementation of elaborated and personalized feedback for the students when they carry out the lists of exercises that the subject's syllabus foresees, thus presenting a progress in the development of the Programming Assistant Introductory plugin (AsPIn). The feedback were validated with students from the University of Brasília in the second half of 2022. In the final survey 87.5% of the group agreed with the effectiveness of the elaborated feedback on their performance in the discipline.

Keywords: Feedback, Introductory programming, CS1, Teaching introductory programming, Moodle, plugin.

Sumário

1	Introdução	1
2	O <i>plugin</i> AsPIn e a disciplina de introdução à computação	3
3	Feedback	4
3.0.1	Feedback no aprendizado inicial em programação	5
3.0.2	Formativo e Motivacional	6
4	Trabalhos relacionados	8
5	Metodologia	10
6	Resultados	16
6.0.1	Perfil dos estudantes	16
6.0.2	Avaliação dos feedbacks	17
7	Análise de resultados	22
8	Limitações e impeditivos	24
9	Conclusão	26
	Referências	28

Lista de Figuras

5.1	Exemplo de um exercício do primeiro módulo da disciplina.	14
5.2	Exemplos de feedbacks no <i>AsPIN</i>	14
5.3	Exemplos de feedbacks no <i>AsPIN</i>	15
5.4	Exemplos de exercícios extras fornecidos no <i>AsPIN</i>	15
6.1	Período que o estudante cursa ICC.	17
6.2	Pesquisa se o aluno já cursou anteriormente ICC.	17
6.3	Conhecimentos prévios de programação.	17
6.4	Estruturação do <i>plugin</i>	18
6.5	Elementos do feedback.	19
6.6	Pesquisa sobre eficácia dos feedbacks em auxiliar o desempenho do aluno. . . .	20
6.7	Pesquisa se o discente notou a presença de exercícios sugeridos junto com vídeo de resolução no final dos feedbacks.	20

Lista de Tabelas

3.1	Estruturação de feedbacks	5
3.2	Composição de um feedback motivacional.	7
5.1	Composição do feedback no <i>plugin</i> AsPIn.	13

Capítulo 1

Introdução

Disciplinas de introdução à programação, desde de 1980, recorrentemente, possuem valores médios acima de 20% de reprovação estudantil, segundo o estudo de Watson e Li [1]. Na obra, apresentam uma pesquisa sobre a alta taxa de reprovação destas disciplinas. É constatado que, dos quinze países abordados, o Brasil representa 1% da amostra coletada e recebe o terceiro lugar no ranking das nações com maior percentagem de reprovação das disciplinas introdutórias à programação, com uma taxa de aprovação de somente 45%.

Feedback é um instrumento habitual na pedagogia e largamente utilizado como ferramenta para um melhor resultado no ensino das linguagens de programação [2]. Com a crescente demanda tecnológica e, conseqüentemente, de profissionais que saibam programar, novas pesquisas são primordiais para o melhor desenvolvimento de capacitações. Assim, um desafio é elaborar feedbacks personalizados, com impacto relevante no aprendizado. Sua personalização e adaptação para diferentes cenários e desempenhos estudantis pode impactar positivamente no ensino básico de programação [3].

A disciplina de introdução à ciência da computação (ICC), da Universidade de Brasília, apresenta ao estudante sua primeira linguagem de programação e compõe a estrutura curricular de cursos como Matemática, Engenharia de Civil, Engenharia Mecânica, entre outros. No total, cada semestre possui uma média de 300 alunos matriculados na disciplina, conforme dados publicamente divulgados pela universidade.

O objetivo deste trabalho é o desenvolvimento de uma série de feedbacks altamente personalizados, com o intuito base de ampliar a percentagem de aprovação na disciplina. Assim, com a inserção do novo recurso, dá-se continuidade no desenvolvimento do *plugin* Assistente de Programação Introdutória, nomeado AsPIn, adaptado ao plano de aula da disciplina Introdução à Ciência da Computação do Departamento de Ciência da Computação da Universidade de Brasília. A disciplina foi escolhida previamente por possuir uma alta taxa de reprovação e permanece no foco da segunda versão do *plugin*.

Este trabalho apresenta, então, uma atualização do *plugin* AsPIn, criado para personalizar a

experiência do estudante ao longo da disciplina ICC, através de uma série de conteúdos adaptados para cada perfil de aluno. Além disso, a atualização busca tornar o aprendizado mais fluido, oferecendo uma série de feedbacks com elementos motivacionais, para que o aluno persista e continue aprendendo, estudando e praticando com motivação e elementos formativos, os quais, de forma resumida, consistem em toda informação que permite ao aluno identificar o que ainda deve ser feito e a forma em que deve ser feita para alcançar os melhores resultados possíveis.

Capítulo 2

O *plugin* AsPIn e a disciplina de introdução à computação

O *plugin* “Assistente de Programação Introdutória”, AsPIn, [4] foi desenvolvido no ano de 2021, em código aberto, com o propósito de auxiliar os alunos no decorrer da disciplina de Introdução à Ciência da Computação, da Universidade de Brasília, identificando o perfil do aluno a partir de um formulário de percepção, no qual são feitas uma série de perguntas para análise de dados educacionais e, a partir de tais dados, gerenciar e fornecer o acesso para o estudante à materiais adicionais de apoio ao aprendizado. O *plugin*, da forma bloco, se apresenta na lateral direita da página da disciplina, no ambiente virtual de aprendizagem *Moodle* [5] e foi desenvolvido na linguagem de programação PHP, semelhante à utilizada para o desenvolvimento do *Moodle*.

Na Universidade de Brasília [6, 7], a disciplina de introdução à computação é ofertada pelo Departamento de Ciências da Computação e está na matriz curricular, como obrigatória, de sete cursos: Matemática - bacharelado (diurno) - 36 vagas; Matemática - licenciatura (diurno) - 36 vagas; Matemática - licenciatura (noturno) - 32 vagas; Engenharia Civil - 40 vagas; Engenharia Elétrica - 40 vagas; Engenharia Florestal- 50 vagas; Engenharia Mecânica - 40 vagas.

O número de vagas por curso se refere a quantidade de vagas disponibilizadas para o ingresso de novos alunos por semestre, totalizando 548 vagas por ano. No primeiro semestre do ano de 2022, período em que foram coletados esses dados, haviam 300 (trezentos) alunos matriculados na disciplina.

Capítulo 3

Feedback

Cohen [8] define feedback como um dos instrumentos com recursos mais poderosos e menos compreendidos no design instrucional e, como explica em sua obra, Black e Wiliam [9] possuem dois principais objetivos, o primeiro é de direcionar, de forma mais específica e precisa, o aluno à revisão e à correção de seu erro. O segundo, é de facilitar o aprendizado de maneira geral, com comentários e sugestões que auxiliam e guiam o aluno na revisão dos conceitos.

Dentro da literatura estudada [10–20], foram encontradas cinco principais classificações de feedback. São elas:

- Motivacional: como o próprio nome já descreve, motiva o estudante a continuar, a seguir em frente, contendo palavras de afirmação.
- Positivo: o qual oferece congratulações pelos resultados.
- Ipsativo: aquele que utiliza dos resultados prévios do próprio estudante, como base e referência, assim, não há comparação com os outros e o desempenho é sempre medido baseado nos próprios resultados passados, incentivando a constante melhoria pessoal.
- Somativo: é avaliado o desempenho do estudante e pode ser simples como uma nota dada.
- Formativo: possui como característica principal, dar continuidade aos ensinamentos passados previamente à avaliação, modificando a forma do estudante de pensar e desenvolver.

Aplicando ao cenário da disciplina de introdução à computação, da Universidade de Brasília, foram, então, desenvolvidos feedbacks personalizados dentro do *plugin* AsPIn. Tal adição, conta com uma estrutura motivacional e formativa, visando somar mais ferramentas para o ensino. Assim, como consequência, maior engajamento estudantil, melhores desempenhos e um aumento do percentual de aprovação são esperados.

3.0.1 Feedback no aprendizado inicial em programação

Dentre as capacidades necessárias para aprender a programar, a solução de problemas e a habilidade matemática são as mais citadas. Além disso, a solução de problemas, bem como a motivação e o engajamento dos alunos, são mencionados como obstáculos para o melhor aprendizado [1–3, 20–22]. De acordo com Dolin *et al* [18], apenas oferecer conceitos ou notas aos estudantes como a única forma de feedback, indicando apenas o desempenho do estudante na avaliação, não é consistente com o objetivo de usar a avaliação para ajudar na aprendizagem.

Narciss [23] define feedback, no escopo de aprendizado relacionado à computação, dividindo em sete grupos baseados em diferentes aspectos do contexto instrucional, como regras de tarefas a serem feitas, erros e conhecimento processual. De maneira sistemática, é possível evidenciar este esquema na Tabela 3.1.

Tabela 3.1: Estruturação de feedbacks

Fatores Instrucionais			
Abordagem Instrucional	Tarefas de Aprendizagem	Erros e obstáculos	Fontes de feedback
- Objetivos e Padrões - Materiais e Métodos - Organização e Comunicação	- Requisito cognitivo - Requisito Meta cognitivo - Requisito motivacional	- Erros conceituais - Erros procedurais - Fonte de erros	- Professor - Parentes - Colegas - Tarefa - Dispositivo técnico
Fatores Individuais			
Objetivos de aprendizagem	Conhecimento e habilidades prévias	Motivação acadêmica	
- Metas de realização do objetivo - Metas de conquista do objetivo	- Erros conceituais - Erros procedurais - Fonte de erros	- Valor de conquistas acadêmicas - Valor intrínscio à tarefa - Expectativas acadêmicas - Habilidades volitivas (de escolha)	
Funções do feedback			
Cognitivo	Meta cognitivo	Motivacional	
Conteúdo do feedback			
Componente avaliativo		Componente informativo	
- Nota, conceito		- Dicas e analogias - Explicações, exemplos, trabalhos, entre outros	
Apresentação do feedback			
Tempo	Sequenciamento	Explicações	Exemplos trabalhados e questões guiadas
Processamento do feedback			
Individual	Social	Motivacional	Cognitivo e meta cognitivo
Efeitos do feedback			
Individual	Social	Motivacional	Cognitivo e meta cognitivo

Assim, referências abordadas e a serem discutidas procuram elucidar a importância e os potenciais benefícios de feedbacks elaborados, assim como formas de estruturá-los e seus desafios. Dolin *et al* [18] concluem, em sua pesquisa, que a maioria dos professores envolvidos consideram possível utilizar exercícios de forma formativa. Desta maneira, a partir dos exercícios adaptados para a matemática, compostos na versão 1.0 do AsPIn [4] e os testes no final dos módulos previstos no plano de ensino da disciplina de introdução à ciência da computação, foram desenvolvidos os feedbacks para essas listas.

3.0.2 Formativo e Motivacional

Shute [12] reúne estudos que sustentam que um feedback formativo efetivo deve incluir elementos de verificação, ou seja, verificar se as respostas estão corretas ou não, e elaboração, que é a parte da mensagem que informa, traz informações relevantes ao aprendizado como um guia para um desempenho de sucesso. Na etapa de elaboração, Shute descreve-a separando em:

1. Abordagem do tema;
2. Abordagem da resposta;
3. Discussão de erros particulares;
4. Apresentação de exemplos trabalhados;
5. Fornecimento de uma orientação gentil;

Cada componente de feedback deve ser atendido, entendido e interpretado e, finalmente, transformado em uma ação corretiva pelo aluno antes que possa influenciar o aprendizado subsequente [23]. Desta forma, o feedback motivacional é utilizado na forma de captar a atenção do aluno, com frases cujas finalidades são encorajar, incentivar e continuar com o bom trabalho, no caso de bons resultados e de não desistir, na ocorrência de um resultado precário. Estão presentes, neste tipo de feedback, funções reforçando os acertos, congratulando o esforço e a persistência, fazendo parte da estratégia do autor de atingir seu objetivo principal de ajudar no aprendizado do estudante. Um feedback motivacional é definido como uma combinação de estratégias motivacionais usando um modelo de motivação com quatro pilares: atenção, relevância, satisfação e confiabilidade. Tal modelo é conhecido como ARSC [24]. Sarsar [17], em sua pesquisa, estrutura um feedback motivacional de acordo com os elementos da Tabela 3.2.

O feedback formativo é considerado eficiente no aperfeiçoamento do conhecimento, em novos aprendizados e na melhoria do desempenho dos estudantes, haja vista que fornece e reforça informações de conteúdo abordado e resolve sobre possíveis falhas do aprendizado [3, 8, 10, 12–16, 20, 23, 25, 26]. Também, é constatada a eficiência da categoria motivacional [17, 24, 27, 28] em detrimento do somativo, visto que este somente informa a nota obtida pelo aluno [18].

É de suma importância observar que há estudos [14] que afirmam que o feedback formativo pode ser prejudicial ao aprendizado, enquanto que, na prática, os estudos e experimentos realizados são sobre dicas automatizadas - simples e diretas - no lugar de um feedback automatizado que visa o aprendizado do aluno, não somente a conclusão da tarefa ou do exercício. Essas dicas automatizadas podem encorajar a tentativa e erro, gerando um abuso sistêmico que raramente ajuda a resolver problemas reais, e que, possivelmente, frustrará o discente.

Tabela 3.2: Composição de um feedback motivacional.

feedback Motivacional		
Estágio	Descrição	Elementos Chave
Atenção	Capturar interesse	<ul style="list-style-type: none"> - Chamar o estudante pelo seu primeiro nome - Usar humor
Relevância	Conhecer as necessidades pessoais e expectativas para uma atitude positiva	<ul style="list-style-type: none"> - Identificar objetivos das tarefas prévias - Descrever a relação entre o objetivo e a atividade - Descrever a relação entre a atividade a área de estudo do aluno
Confiança	Consciência de alcançar e controlar o sucesso	<ul style="list-style-type: none"> - Confirmação de que as expectativas foram atingidas - Fornecer recomendações para aperfeiçoar as habilidades - Fornecer informações sobre qualidade e correção (se necessário)
Satisfação	Reforço	<ul style="list-style-type: none"> - Fornecer uma nota para o teste - Elogiar os esforços do aluno

Capítulo 4

Trabalhos relacionados

Phye e Sanders [11] realizaram dois experimentos sobre resolução de problemas, mapeando índice de acertos e tipos de erros, com dois tipos de feedback. São eles: genérico, com informações e recomendações gerais, e específico, com um maior número de informações sobre o conteúdo e, também, de acordo com o desempenho do estudante. Esses experimentos concluem que o tipo de feedback mais específico é superior ao genérico. As taxas de erro iguais (ao longo de ambos experimentos) diminuem significativamente, assim como a proporção de erros novos. Também é notado que há uma maior retenção dos voluntários que deveriam resolver a série de tarefas de resolução de problemas.

Miller [29], por sua vez, realizou três experimentos com enfoque em feedback formativo, em avaliações baseadas em computador. No fim dos experimentos, foram coletados feedbacks dos alunos que, apesar de serem somente 2.5% da população, reportaram uma experiência bastante favorável ao utilizar a ferramenta, o que indicou uma alta taxa de aprovação com sua aplicação. Meses depois, foi realizada uma nova pesquisa entre os alunos participantes, com uma taxa de resposta bem mais relevante, de 70%. Foi concluído que direcionamento dos alunos a materiais de apoio é um feedback efetivo e que os alunos indicaram que estes foram altamente eficazes no apoio à sua aprendizagem.

Com o objetivo similar à pesquisa anterior, ou seja, de promover um aumento na curva de aprendizado utilizando feedbacks elaborados, Silva, Costa e Araújo [3] utilizaram um metodologia de promover um feedback formativo durante os exercícios de programação. O aluno deveria pedir ajuda selecionando ou não uma parte do código. Desta maneira, o algoritmo busca em sua base e apresenta um feedback mais próximo à ajuda necessária ao estudante. O experimento realizado possui como resultado um aumento no desempenho nos exercícios, após a introdução dessa ajuda dinâmica e em tempo real.

Hao e Tsikerdekis [14] desenvolveram um microserviço para fornecer feedbacks formativos para alunos em suas tarefas de programação. Ademais, foi criada uma integração com o repositório para o qual os estudantes enviavam seus códigos e, através deste, o microserviço

examinava os trabalhos submetidos e enviava um feedback para o estudante. Obtiveram como resultado: a aprovação dos estudantes, os quais sentiram motivados a continuar realizando os exercícios, após receberem os feedbacks e a confirmação das descobertas sobre a relação positiva entre a busca de feedback formativo e realizações acadêmicas.

Na pesquisa de Meier e Lepp [20] sobre a eficácia do feedback em cursos de introdução à programação, para alunos que não são de cursos relacionados à computação, é constatado que feedbacks, na maioria dos cursos, são somente fornecidos aos alunos nos resultados finais do curso, atraindo pouca atenção ao processo de programação em si. O experimento apresentou dados que mostram que o grupo que recebeu os feedbacks adicionais recebeu notas maiores e desempenho do código melhor em relação ao tempo de conclusão, número de execuções e mensagens de erro quando comparado ao grupo que cursou a disciplina no mesmo momento, sem interferência do experimento.

Diferentemente dos artigos apresentados, a proposta deste trabalho é de um sistema aberto de feedback executado no ambiente *moodle*, possibilitando sua aplicação em outras diversas universidades. Entretanto, a ferramenta foi construída com foco no funcionamento da disciplina de introdução à ciência da computação da Universidade de Brasília. Foi utilizada uma integração de diferentes componentes motivacionais e formativos no feedback, tais como motivações com mensagens personalizadas, dicas e informações acerca do conteúdo, além de sugestão de exercícios complementares e desafios no escopo do conteúdo do módulo atual ou do seguinte. Porém, a ferramenta foi aplicada em uma estrutura menos formal, com o objetivo de se aproximar mais do aluno, à medida que este avança na disciplina e, por consequência, obter uma maior aceitação.

Capítulo 5

Metodologia

Este estudo possui como objetivo a elaboração de feedbacks personalizados, formados pela junção dos elementos característicos motivacionais e formativos, em uma configuração em que cada aluno, de acordo com seu desempenho em cada teste realizado, receba um feedback específico. Desta forma, ajudando-o a melhorar ou manter um ótimo desempenho, assim como motivar a não desistir ou até mesmo ir além do conhecimento básico necessário para concluir o módulo estudado.

Os feedbacks foram elaborados e inseridos no código fonte do *plugin* ASPIn, na linguagem de programação PHP. O desenvolvimento foi feito por meio da criação de quatro funções que resultaram na apresentação apropriada do feedback personalizado ao usuário, à medida este conclui cada módulo, termina a lista de exercícios e avança na disciplina.

A primeira função desenvolvida aloca dinamicamente em uma variável a nota que o discente recebeu em determinada lista. A segunda função foi criada para aprimorar o código já existente da primeira versão [4], em que era necessário o prazo da lista terminar para obter as notas e, assim, enviar um feedback. Com esta função, basta o aluno finalizar a lista que já recebe seu feedback personalizado. A terceira função obtém-se do *moodle*, o primeiro nome do aluno, utilizando-o na introdução de cada texto. Por fim, foi criada a função de feedback por nota e por lista, por meio da qual é feita uma verificação de qual módulo o aluno se encontra e qual foi a nota tirada e retorna o feedback adequado.

Nesta última função, fica localizado todo o texto elaborado e suas possibilidades de apresentação ao discente. Para o estudante receber o feedback personalizado, foram desenvolvidas duas condicionais. Primeiramente, verifica-se qual foi a última lista que o estudante concluiu, por meio da nota. E, em seguida, verifica qual menção o discente obteve para, assim, fornecer o feedback correto.

Os feedbacks foram desenvolvidos para cada um dos módulos da disciplina e foram agrupados em quatro grupos: feedback para alunos que tiraram nota menor que a média 5, feedback para resultados de 5 até 5,9, outro para resultados entre 6 e 7,9 e, finalmente, para o ótimo re-

sultado de 8 até a nota máxima, 10. Cada feedback, com proposta de exercícios extras e uma resolução para cada um deles.

Os feedbacks foram então modelados, baseados em etapas de Narciss [23]. A primeira é comunicação dos critérios de forma a ajudar os alunos a concluírem cada uma das listas desenvolvendo exatamente o que cada exercício pede e a forma que o corretor espera a resposta. Essa primeira etapa já se encontra nas próprias listas, como no exemplo da Figura 5.1.

A segunda e terceira etapas de Narciss remetem ao autoconhecimento, feedback interno e avaliação do nível de desempenho, todas elas por parte do aluno. Dessa forma, não foram aplicadas, visto que o objetivo do trabalho é focado em feedbacks formativos e motivacionais no ensino introdutório à computação. A quarta etapa de Narciss é iniciar a reflexão sobre ações corretivas para fechar lacunas entre o nível atual de desempenho e os padrões. A quinta, é fornecer informações de tutoria, com dicas, perguntas de orientação, explicações, analogias para ajudar os alunos a melhorarem seu desempenho. Sexta etapa é oferecer novas formas de envolver o aluno em ações que aplicam as informações de feedback de tutoria e, por fim, a sétima é provocar progressos e enfatizar realizações bem-sucedidas de alta qualidade e padrões, uma espécie de feedback motivacional.

A elaboração do feedback, no *plugin*, foi realizada baseada também nos dois componentes principais de Shute [12], verificação e elaboração. A verificação foi feita baseando-se nas menções que a Universidade de Brasília utiliza. Portanto, para cada módulo do curso, foram criadas quatro variações de feedback. Para notas inferiores à média 5 de 10 pontos, é apresentado para o usuário um feedback mais elaborado, vide Figura 5.2a, sendo a primeira parte, motivacional, com um potencial objetivo de torná-lo mais pessoal e mais motivacional, aumentando assim, sua eficiência [17]. Logo após, discute-se possíveis erros cometidos na escrita do programa, apresenta-se diversos exemplos específicos do tópico estudado em tal módulo, como também são mostradas soluções de questões similares às exigidas no teste e, por fim, um vídeo é indicado por questão, de um professor que guia, passo a passo, os espectadores à resolução dos exercícios sugeridos, explicando a utilização de todas as funções, variáveis e sugestões de boa prática na elaboração do programa.

Para uma nota entre 5,0 e 6,9, é apresentado um feedback que difere-se do anterior, iniciando com uma congratulação ao estudante, a fim de confirmar que as expectativas foram atingidas, porém trazendo elementos que afirmam o potencial do aluno, incentivando-o a melhorar. Confira um exemplo na Figura 5.2b. Assim, são fornecidos, de forma semelhante à anterior, os elementos instrucionais, seguido de exercícios sugeridos e explicações do tema abordado e resolução.

Para o desempenho entre 7,0 e 8,9, vide Figura 5.3a. O feedback possui um tamanho mais reduzido, porém ainda contendo todos os elementos. Isto é, saudação com o nome do estudante, parabenização pelo resultado, de forma mais intensa à anterior, por ter obtido uma nota mais

satisfatória, apresentação de possíveis erros mínimos que levaram o *software* de avaliação à descontar algum pequenos pontos, como, por exemplo, não abordar todos os casos e um teste ter falhado, em alguns exercícios. E, por último, uma indicação de desafios com questões que exigem uma maior complexidade na formulação dos algoritmos daquele conteúdo estudado e desafios que abordam o conteúdo a ser estudo no módulo seguinte.

Para notas entre 9,0 e o máximo, 10, o feedback aborda aspectos muitos semelhantes ao anterior, conforme o exemplo da Figura 5.3b, com mais ênfase na congratulação do estudante, por ter obtido um desempenho excelente. Em seguida, são apresentadas informações de utilização dos tópicos cobrados no módulo em questão, com uma sugestão de vídeo educativo e desafios sobre este e o próximo tópico a ser estudado, conforme plano de ensino da disciplina. Também é fornecido um vídeo para cada desafio, com o exemplo trabalhado, com explicação e dicas. Este é o feedback mais sucinto, objetivando apenas congratular o aluno e incentivá-lo a continuar com o excelente resultado. Todavia, se o aluno possuir grande facilidade no conteúdo, é importante fornecer uma variedade de opções que faça-o se sentir estimulado e interessado [28], da mesma forma que são oferecidos diversos exercícios, de diferentes complexidades e inteiramente opcionais, para estudantes de outros desempenhos.

Shute [12] apresenta e define em sua obra que feedbacks específicos e elaborados são, de forma geral, melhores, em termos de eficiência, que os mais genéricos. Todavia, se os mais elaborados forem muito longos ou muito complicados, muitos estudantes podem, simplesmente, não engajarem na leitura ou não prestarem a atenção necessária nas considerações desenvolvidas, tornando os feedbacks sem valia.

Kulhavy [15] realizou experimentos em relação ao tamanho ideal de um feedback. Entretanto, por ter obtido uma gama muito variada de dados em sua apuração, conclui-se que resultados foram inconsistentes. Diante dos elementos expostos, o feedback para o AsPIn foi construído com os elementos presentes na Tabela 5.1. Em seguida, nas Figuras 5.4a e 5.4b, manifestam-se exemplos de exercícios de acordo com o desempenho do estudante. Na primeira imagem, exercícios mais simples, aplicados à matemática como um reforço ao conteúdo estudado no módulo e, na segunda, desafios para alunos com um desempenho superior.

Tabela 5.1: Composição do feedback no *plugin* AsPIn.

Composição do feedback no AsPIn		
Componente	Objetivo	Descrição
Introduz chamando pelo nome	<ul style="list-style-type: none"> - Capturar atenção - Personalização - Aproximação e intimidade 	<ul style="list-style-type: none"> - Captura a atenção do professor para a configuração inicial do plugin - Captura a atenção do estudante para o feedback e às atividades a serem feitas
Motivação	<ul style="list-style-type: none"> - Aumentar/manter o interesse - Demonstrar reconhecimento do esforço - Demonstrar orgulho - Demonstrar que o desempenho individual do aluno é importante 	<ul style="list-style-type: none"> - Congratular pelo desempenho - Descrever a relação entre a atividade a área de estudo do aluno - Motivar a melhorar a nota - Motivar a continuar se esforçando - Motivar a treinar mais - Motivar a ir além (com problemas desafios)
Formação	<ul style="list-style-type: none"> - Ressaltar a forma que o sistema avalia os programas - Provocar uma reflexão sobre ações corretivas - Fornecer informações de tutoria e dicas - Promover a prática contínua - Fornecer exercícios extras - Fornecer desafios - Fornecer resultados esperados dos exercícios - Fornecer um passo a passo didático 	<ul style="list-style-type: none"> - Exemplificar o que é esperado do aluno e quais são os critérios avaliados pelo corretor - Exemplificar os principais erros de programação, como lógica e digitação - Sugerir em como o aluno pode melhorar e informar que os professores e monitores estão sempre à disposição para sanar as dúvidas - Informar que o conhecimento em programação é desenvolvido e aprimorado a partir da prática e do treinamento - Apresentar exercícios extras, em que o principal objetivo é justamente aquele que foi abordado e cobrado no módulo atual estudado, apresentando entradas e saídas exemplificando a forma que o programa deve funcionar - Apresentar desafios para aqueles alunos com bons desempenhos, para que se mantenham estimulados à prática, sugerindo tanto desafios com o conteúdo da próxima seção a ser estudada, como desafios com o conteúdo do módulo atual, porém com um nível de dificuldade mais elevado, levando o estudante a desenvolver um programa mais elaborado. - Apresentar um vídeo de cada questão fornecida, com uma explicação de um professor sobre o conteúdo em questão estudado, discussão da questão, o pensamento computacional e lógica para resolvê-la e, por fim, a elaboração completa de uma solução para tal.

Comentários:

Para ler vários valores em uma mesma linha, use `input().split()`. Se o argumento de `split` for vazio, o separador das variáveis é um espaço em branco. Porém, lembre-se que `input` lê apenas strings do teclado, portanto você deverá converter as strings em floats. No exemplo a seguir, o usuário digita valores separados por um espaço em branco e aperta enter para enviá-los, então, o programa lê esses valores separados por espaços como strings (na ordem em que aparecem), guardados nas variáveis correspondentes. Em seguida, o programa os converte para floats:

```
A, B, C = input().split()
A, B, C = [float(A), float(B), float(C)]
```

Números complexos podem ser manipulados facilmente com o método `complex()`. Exemplo:

```
D = complex('2+3j')
print("%.2f" % D.real)
print("%.2f" % D.imag)
```

Dica:

O módulo `math` pode ser usado para calcular a raiz quadrada com a função `math.sqrt(A)`. Neste caso, a biblioteca precisa ser importada antes: `import math`.

Existe um operador `**` em Python que faz a exponenciação. Exemplo: `A ** B = AB`

For example:

Input	Result
1.0 7.0	4.4721
5.0 9.0	2.0000
2j	
-2.5 0.4	16.1484
12.1 7.3	2.2361
1+2j	

Figura 5.1: Exemplo de um exercício do primeiro módulo da disciplina.

Admin, infelizmente, você não atingiu a média neste módulo, mas não desista! Programar pode ser uma tarefa complicada e estamos aqui para te ajudar. Para isso, selecionamos exercícios opcionais para você treinar estruturas de decisão, o laço "for" e poder assistir como um professor resolve esses problemas. A forma de escrever o código, por onde começar, como entender o problema e o passo a passo para você conseguir se sair melhor nos próximos módulos! Dê uma conferida [nessa aula](#) sobre a estrutura de repetição "while". Os vídeos, ensinam uma formato de pensamento computacional que será bastante útil para desenvolver seus programas na linguagem e fará com você, exercícios aplicando a nova estrutura aprendida neste terceiro módulo do curso. Assista o vídeo a seguir com alguns exercícios para praticar e confira [aqui](#) a resolução da professora.

Para praticar mais e se sair melhor nas próximas vezes, aqui estão dois problemas propostos, opcionais que irão te mostrar de maneira prática, como evitar grandes repetições e solucionar problemas utilizando uma Estrutura de Repetição (while), diferente da aprendida anteriormente. Tente você primeiro, antes de assistir a solução!

Não desista, lembre-se que os professores e monitores estão à sua disposição para te ajudar em por todo o caminho no aprendizado de programação.

Admin, parabéns, você passou na lista 3, mas ainda há espaço para melhorias! Não deixe de verificar os detalhes que errou, atente-se à especificação para que seus códigos estejam conforme o padrão especificado nos exemplos de cada questão, muitas vezes, é só um ajuste na saída para que o corretor automático corrija seu código com sucesso e sem penalidades. Para se sair ainda melhor nos módulos que vêm pela frente, vou te indicar alguns exercícios extras de estrutura de decisão e laço for para você treinar mais. Mas antes, dê uma conferida [nessa aula](#) sobre a estrutura de repetição while.

O vídeo, ensina uma formato de pensamento computacional que será bastante útil para desenvolver seus programas na linguagem e fará com você, exercícios aplicando a nova estrutura de laço aprendida neste terceiro módulo do curso.

Assista [este vídeo](#) com alguns exercícios para praticar e confira a resolução da professora. Os problemas propostos a seguir, irão te mostrar de maneira prática, como evitar grandes repetições e solucionar problemas utilizando uma Estrutura de Repetição (while) diferente da aprendida anteriormente. Tente você primeiro, antes de ver a solução.

(a) Abaixo da média;

(b) Média;

Figura 5.2: Exemplos de feedbacks no *AsPIn*.

Admin, parabéns, você obteve um resultado muito bom na lista 3. Não deixe de verificar os detalhes que errou. Atente-se à especificação para que seus códigos estejam em conformidade com o padrão especificado nos exemplos de cada questão, muitas vezes, é só um ajuste na saída para que o corretor automático corrija seu código com sucesso e sem penalidades. Procure realizar os exercícios propostos de forma mais concisa possível, ou seja, somente o básico para imprimir o que é pedido em cada exercício proposto. Mas não desanime, sua nota foi muito boa. Se desejar continuar aprimorando seu conhecimento não deixe de conferir [este vídeo](#) sobre interrupção dos laços "while", como utilizar a instrução "break" e os "loopings" infinitos a favor de estratégias de seu código e exemplos práticos que serão utilizados e a necessidade de utilizar uma quebra no meio do laço. Além disso, também será mostrado um pouquinho de fstrings na linguagem.

Aproveite e não se esqueça: os professores e monitores estão à sua disposição para te ajudar em suas dúvidas e propor novos desafios. Até o próximo módulo!

Admin, parabéns! Você concluiu a lista 3 com um ótimo resultado! Continue se esforçando para manter o excelente rendimento em todo o curso. Para se aprofundar ainda mais, existem características específicas dos laços de repetição que podem causar erros graves em nossos programas ou, se souber utilizar, podem ser grandes aliados em nossos códigos. Veja este ótimo [vídeo](#) sobre interrupção dos laços "while", como utilizar a instrução "break" e os "loopings" infinitos a favor de estratégias de seu código e exemplos práticos que serão utilizados e a necessidade de utilizar uma quebra no meio do laço. Além disso, também será mostrado um pouquinho de fstrings na linguagem.

Aproveite e não se esqueça: os professores e monitores estão à sua disposição para te ajudar em suas dúvidas e propor novos desafios. Até o próximo módulo!

(a) Média superior;

(b) Superior;

Figura 5.3: Exemplos de feedbacks no *AsPIn*.

Ex. 1: Tratando vários valores
Crie um programa que leia vários números inteiros pelo teclado. O programa só vai parar quando o usuário digitar o valor 999, que é a condição de parada. No final, mostre quantos números foram digitados e qual foi a soma entre eles (desconsiderando o flag).

Input	Result
7, 6, 10, 999	Números: 3, Soma: 23
4, 999	Números: 1, Soma: 4
999	Números: 0, Soma: 0
888, 1, 999	Números: 2, Soma: 889

Solução passo a passo

Ex. 2: Tabuada usando while
Faça um programa que mostre a tabuada de vários números, um de cada vez, para cada valor digitado pelo usuário. O programa será interrompido quando o número solicitado for negativo. Obrigatório: usar um laço de repetição while, aprendido neste módulo.

Input	Result
	2 x 1 = 2
	3 x 2 = 6
	2 x 3 = 6
3	3 x 4 = 12
	2 x 5 = 10
	3 x 6 = 18
	3 x 7 = 21
	3 x 8 = 24
	3 x 9 = 27
	3 x 10 = 30

Solução passo a passo

(a) Exercícios sugeridos;

Desafio 1: Detector de Palíndromo
Crie um programa que leia uma frase qualquer e diga se ela é um palíndromo (palavra que se pode ler, indiferentemente, da esquerda para a direita ou vice-versa), desconsiderando os espaços, vírgulas, hifens e quaisquer outras pontuações, acentuações e diferença entre letras maiúsculas e minúsculas.

Input	Result
Mariam, Tm, Admin,	é um palíndromo
Cia, sou um programador	Não é um palíndromo
Socorro-me, não se deslize em fofocas	é um palíndromo
Ada e topa	é um palíndromo

Solução passo a passo

Desafio 2: Analisador completo de pessoas
Desenvolva um programa que leia o nome, idade e sexo de 4 pessoas. No final do programa, mostre: a média de idade do grupo, qual é o nome do indivíduo do sexo masculino mais velho e quantos indivíduos do sexo feminino possuem menos de 20 anos.

Input	Result
Artur, 20, Masculino Amanda, 18, Feminino Clara, 19, Feminino Roberto, 15, Masculino	A média de idade é 19 anos Artur é o mais velho 2 indivíduos

Obs: Quer tornar mais interessante?
Altere a entrada para receber a data de nascimento, ao invés da idade e faça seu programa calcular a idade dos indivíduos.

Input	Result
Artur, 20/10/2002, Masculino Amanda, 18/07/2004, Feminino Clara, 19/02/2003, Feminino Roberto, 05/04/2005, Masculino	A média de idade é 19 anos Artur é o mais velho 2 indivíduos

Solução passo a passo

(b) Desafios sugeridos;

Figura 5.4: Exemplos de exercícios extras fornecidos no *AsPIn*.

Capítulo 6

Resultados

Com a disponibilização do ambiente de testes “Aprender2”, cedido pela Universidade de Brasília, foi possível simular os dois primeiros módulos da disciplina introdutória de programação. No segundo semestre de 2022, momento no qual o experimento foi realizado, a disciplina possuía um total de 276 alunos matriculados. 110 discentes se inscreveram para participar, resultando em aproximadamente 40% dos estudantes de ICC. No final do prazo estipulado, foram obtidas 32 respostas, ou seja, 29,1% do número de intenções e 11,6% do total de discentes matriculados.

6.0.1 Perfil dos estudantes

A primeira seção da pesquisa possui como objetivo perfilar os alunos quanto ao período em que cursa a disciplina de introdução à computação, se esta é a primeira vez cursando e, caso não seja, se trancou ou reprovou. Adicionalmente, é perguntado se o pesquisado possui conhecimentos prévios em programação. De acordo com a Figura 6.1, é possível observar que 28,125% cursam a disciplina no primeiro semestre, 25% no segundo e 46,875% cursam-a no terceiro semestre em diante. A Figura 6.2 evidencia o percentual de alunos que estão cursando a disciplina pela primeira vez ou se já cursaram anteriormente. A maioria, 81,25%, está fazendo pela primeira vez, 12,5% reprovou e 6,25% trancou ou retirou.

A Figura 6.3, referente a conhecimentos prévios de programação, indica que a maioria dos discentes nunca antes tinham programado e, em contrapartida, apenas quatro indicaram que já sabiam programar. Isso reforça o formato que os feedbacks foram construídos, com enfoque no estudo introdutório de programação. Finalizando a seção de perfilamento, é perguntado qual é o curso de graduação dos pesquisados. O resultado obtido foi: 21,875% de estatística, em seguida, os cursos de engenharia mecânica, física e matemática, todos com 15,625%, engenharia de produção com 12,5% e o restante, com o mesmo número de discentes, administração,

educação física, engenharia civil, engenharia florestal, engenharia mecatrônica e fisioterapia, representando 3,125% cada um.

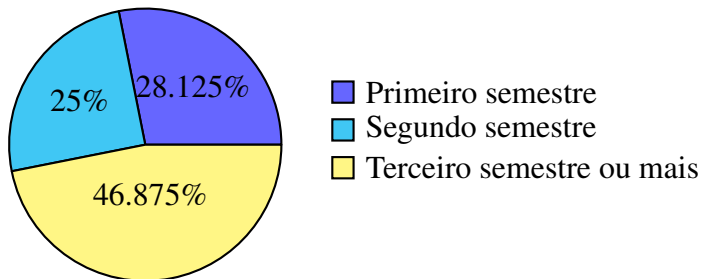


Figura 6.1: Período que o estudante cursa ICC.

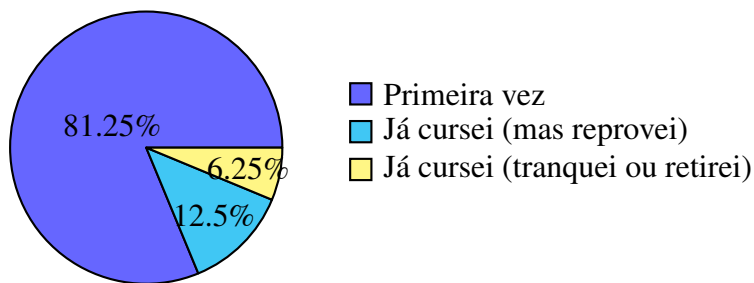


Figura 6.2: Pesquisa se o aluno já cursou anteriormente ICC.

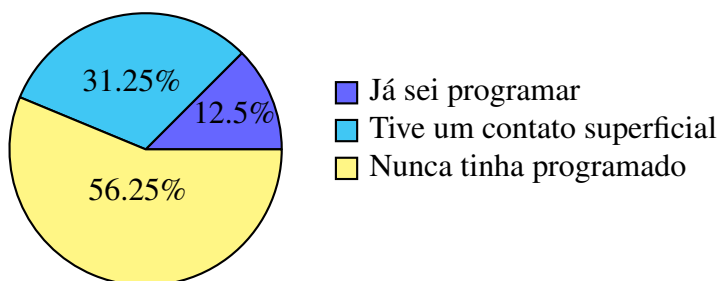


Figura 6.3: Conhecimentos prévios de programação.

6.0.2 Avaliação dos feedbacks

Por fim, a segunda seção da pesquisa é uma avaliação dos estudantes, em relação aos feedbacks. A Figura 6.4 apresenta o resultado de uma pesquisa de opinião, em escala Likert [30], na qual foi realizada uma série de afirmações sobre os elementos do AsPIIn e os estudantes especificam seu nível de concordância.

A primeira afirmação é: "Receber uma mensagem me lembrando da importância das listas de exercício me motiva a realizá-las." 43.8% concordam totalmente, 37.5% concordam parcial-

mente e 18.8% são indiferentes. A segunda afirmação, “Receber uma mensagem de parabenização por ter ido bem na lista de exercícios me motiva a continuar estudando.”, 71.9% concordam totalmente, 18.8% concordam parcialmente e 9.4% são indiferentes. Em seguida, “Receber uma mensagem de motivação ao ter ido mal em uma lista de exercícios me ajudaria a perseverar na disciplina”, 46.9% concordam totalmente, 40.6% concordam parcialmente e 12.5% são indiferentes. O restante é possível conferir com detalhes na Figura 6.4.

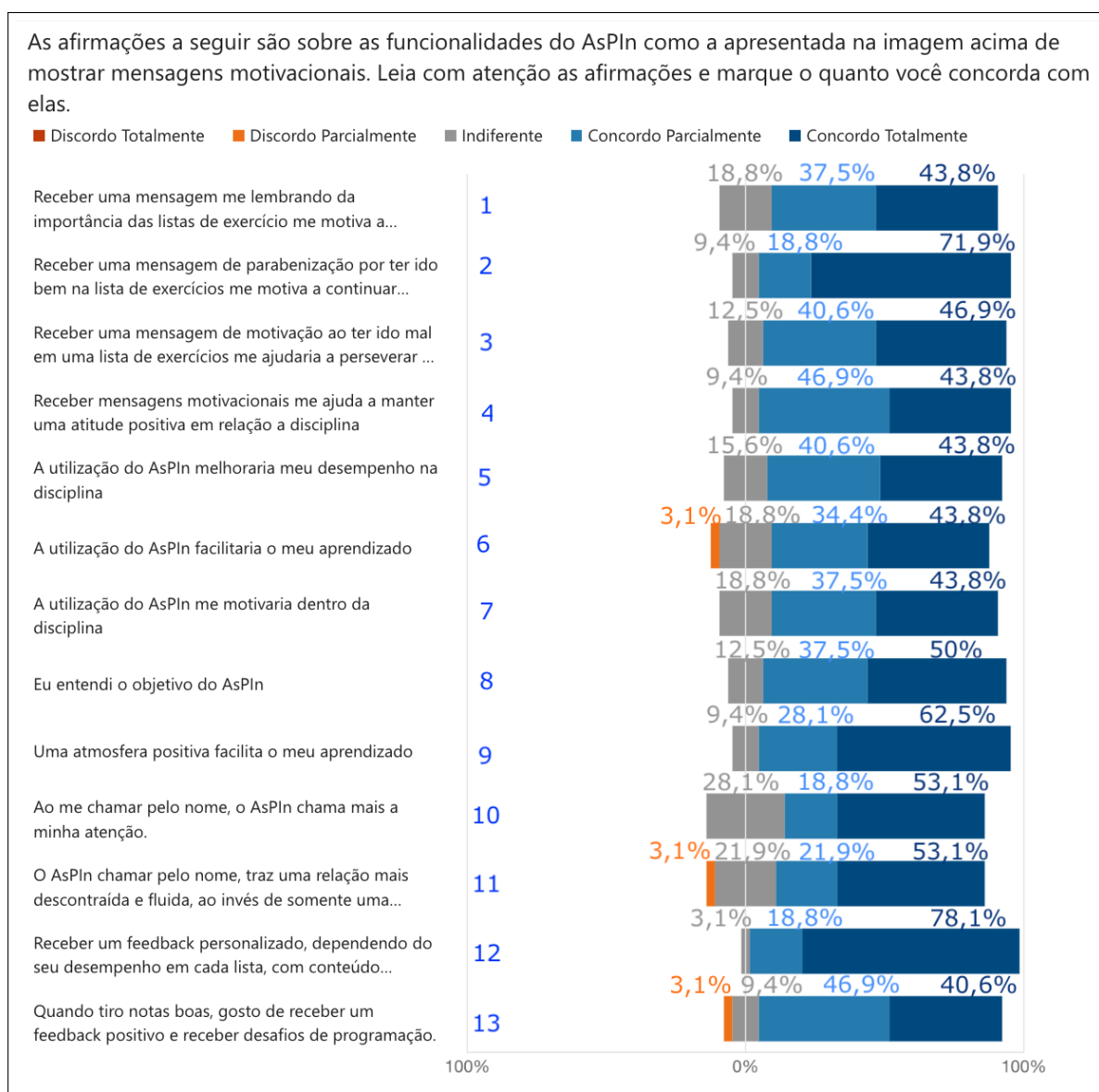


Figura 6.4: Estruturação do *plugin*.

Na Figura 6.5, também em escala *Likert*, são apresentados elementos específicos formativos e motivacionais que compõem o feedback e, em uma escala de 1 a 5, obtem-se o nível de interesse que os discentes possuem por tais elementos. Na primeira afirmação é dito: “Presença de elementos motivacionais para motivar os alunos, com uma menor nota, a continuarem se es-

forçando.", 62.5% acham muito interessante, 31.3% acham parcialmente interessante, 3.1% são indiferentes e 3.1% acham pouco interessante. No número 2, "Presença de elementos motivacionais para congratular os alunos que obtiveram boas notas, demonstrando reconhecimento.", 62.5% acham muito interessante, 28.1% acham parcialmente interessante, 6.3% são indiferentes e 3.1% acham pouco interessante. Já na quarta afirmação, em "Presença de elementos formativos. Isto é, dicas para realizar exercícios, mostrar erros mais comuns ao escrever um programa e um conteúdo adicional educativo.", 84.4% acham muito interessante, 12.5% acham parcialmente interessante e 3.1% são indiferentes.

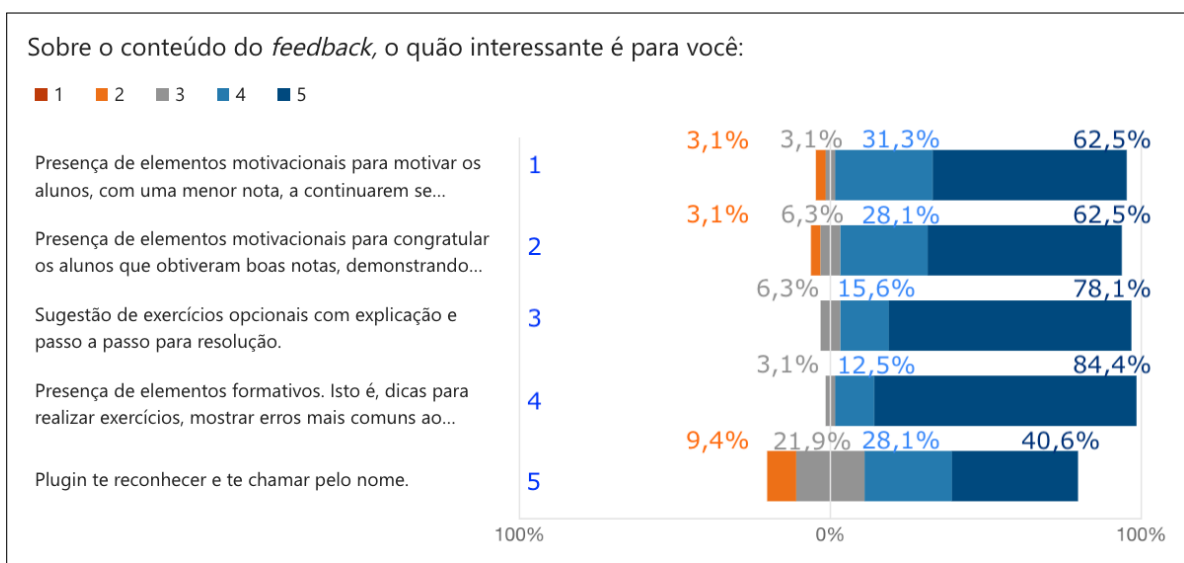


Figura 6.5: Elementos do feedback.

Em seguida, foi indagado aos alunos se acreditam na eficácia dos feedbacks no auxílio de seu desempenho. O gráfico na Figura 6.6 indica uma resposta positiva, com 87,5% de concordância no emprego proveitoso da funcionalidade. Logo depois, quando perguntados, 27 dos 32 estudantes, totalizando em 84,375%, afirmaram terem visto os exercícios sugeridos e os vídeos resolutivos, vide Figura 6.7. O que, por serem os últimos elementos dos feedbacks, sugere que a maioria dos alunos, de fato, percorreram todo o texto, demonstrando interesse no conteúdo desenvolvido.

Quando perguntados, 27 dos 32 estudantes afirmaram terem visto os exercícios sugeridos e os vídeos resolutivos, vide Figura 6.7.

Por fim, a pesquisa possuía dois campos texto opcionais. O primeiro, era para que os discentes pudessem escrever livremente sugestões e comentários sobre os *feedbacks personalizados* e o preenchimento desta etapa obteve um índice de participação de 43,75%. Os comentários foram todos positivos, com elogios à abordagem motivacional e às informações adicionais sobre a matéria. As sugestões foram:

- Mais detalhamento de quais foram os erros cometidos no último código enviado.

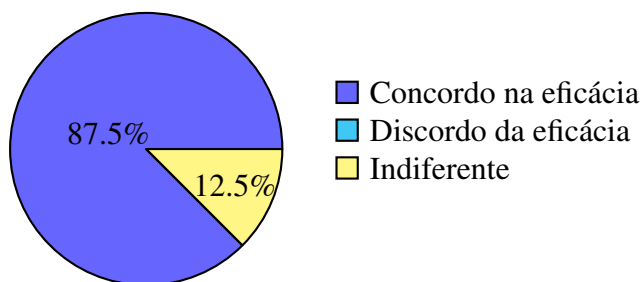


Figura 6.6: Pesquisa sobre eficácia dos feedbacks em auxiliar o desempenho do aluno.

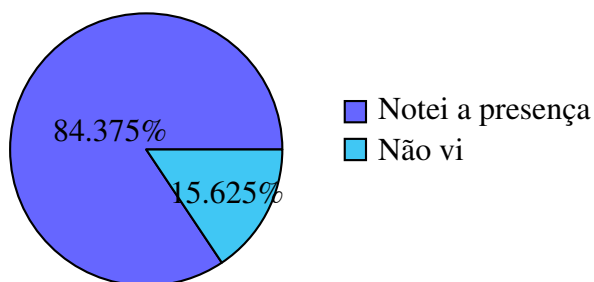


Figura 6.7: Pesquisa se o discente notou a presença de exercícios sugeridos junto com vídeo de resolução no final dos feedbacks.

- Feedbacks descontraídos, surpresos ou impressionados com desempenhos excelentes.
- Mais exercícios extras.
- Ao avisar sobre a próxima lista, apresentar o próximo conteúdo com uma aplicação e uma breve explicação do ponto mais importante da seção.
- Melhorar a visibilidade do *plugin*.
- Gostaria de receber mais informações quando minhas respostas estão parcialmente corretas.

O segundo campo texto era para, especificamente, os estudantes relatarem suas opiniões acerca dos exercícios e desafios sugeridos. Foi obtida uma participação de 62,5%. As principais respostas foram:

- “Achei que foram importantes e interessantes e acho que possuem um papel importante sim, quanto mais prática, mais fixação.”
- “Gostei bastante, pois demonstram em quais instâncias seria bom utilizar certas estruturas e operadores. Acredito que sem esses exercícios extras, o conteúdo que eu aprendi seria facilmente esquecido.”
- “Gostei, é importante a inserção de uma variedade maior de exercícios ou um maior grau de dificuldade, mesmo que opcionais, para proporcionar ao estudante uma oportunidade de aprofundamento.”

- “São interessantes. É mais uma forma de ver e saber como podemos aplicar as estruturas que estamos estudando.”

Capítulo 7

Análise de resultados

O perfilamento dos pesquisados indica que, de fato, mesmo havendo 46,875% de alunos a partir do terceiro semestre de curso, vide Figura 6.2, 56,25% não possuem conhecimentos prévios de programação, indicado na Figura 6.3, justamente o foco principal da ferramenta desenvolvida neste projeto. Como indicado pela Figura 6.1, apenas 37,5% estão cursando ICC como disciplina obrigatória, isto é, os alunos da matemática, engenharia mecânica, engenharia florestal e engenharia civil. Apesar da disciplina possuir uma alta taxa de reprovação, apenas 12,5% dos estudantes indicaram que já tinham reprovado, sendo apenas 3,125% com a disciplina na matriz curricular do curso.

A escala *Likert* sobre a Estruturação dos feedbacks 6.4, indica uma alta taxa de aprovação geral da metodologia utilizada. Obteve-se um média geral de 84,88% de marcações em que os estudantes concordam com as afirmações a respeito, além de não haver nenhum que discordou totalmente com qualquer uma das afirmativas. Na segunda escala 6.5, o resultado foi ainda mais positivo, com uma aprovação média de 90,1% sobre os elementos dos feedbacks e também não houve nenhuma marcação de menor nota. Sendo assim, todos os elementos, com exceção de chamar o aluno pelo nome, obtiveram uma taxa de aprovação maior que 90%. Observa-se, também, na Figura 6.6, a alta taxa de concordância, 87,5%, sobre a eficácia dos feedbacks.

Sobre a Figura 6.3, a maioria dos alunos tende a possuir mais dificuldade ao longo da disciplina, por nunca terem programado anteriormente. Destes, 83,33% afirmam que ao terminar cada lista de exercício, os feedbacks recebidos podem auxiliar em seu desempenho, além de que 61,11% concordam totalmente que a sua utilização e a atmosfera positiva do conteúdo facilitaria o aprendizado, enquanto 27,78% concordam parcialmente e o restante, 11,11%, são indiferentes.

Analisando a Figura 6.7, pode-se concluir, por os exercícios sugeridos junto com vídeo de resolução serem os últimos elementos dos feedbacks, que os alunos percorreram todo o texto, demonstrando interesse no conteúdo desenvolvido. Também, indica que os feedbacks não estão com um tamanho excessivo. Tal estrutura e tamanho foi questionado, contrastando

com a importância de fornecer aos discentes, uma variedade de opções. Pois, utilizando uma estratégia de fornecer uma maior e diversa quantidade de informações e, conseqüentemente, formas de feedback, cria uma sobrecarga de informações. Além de um ambiente confuso para muitos discentes, fazendo com que estes desistam de usar ou não cheguem até o fim [25].

Em relação aos comentários nos campos texto, foram obtidas uma série de sugestões e críticas pertinentes, além de opiniões bastante positivas acerca do projeto desenvolvido, abrindo uma série de opções de novas funcionalidades a serem desenvolvidas em trabalhos futuros. Todavia, o número de alunos que voluntariam-se à realizar a pesquisa e responder o questionário avaliativo pode limitar o poder dos dados estatísticos coletados.

Capítulo 8

Limitações e impeditivos

É necessário considerar os fatos que limitaram o desenvolvimento dos feedbacks para a versão 2.0 do *plugin* AsPIn. O *plugin* foi criado e aprimorado para a disciplina de introdução à ciência da computação da Universidade de Brasília, o que depende da estrutura e da forma como a disciplina é ministrada. Inicialmente, o *plugin* foi criado na época da pandemia do COVID-19 e, sua versão 2.0, foi desenvolvida no período de transição entre os semestres em que as aulas eram ministradas de forma integralmente remotas, por meio de video conferências e utilização total do ambiente Aprender. Porém, no segundo semestre do ano de 2022, as aulas retornaram, completamente, para a estrutura original presencial.

Suplementar aos fatores expostos, houve também a preocupação quanto ao tempo e motivação dos alunos voluntários para realizarem os testes no ambiente com o *plugin*. Dada a extensão do curso, com sete listas de exercícios, torna-se inviável reproduzir a disciplina em toda sua extensão para realizar uma experimentação com os matriculados por dois principais motivos, o primeiro é justamente atrair voluntários para realizarem um curso por completo, sem no final serem habilitados com o certificado de conclusão de tal disciplina; o segundo, é não ter tempo hábil para realizar um experimento tão extenso como é previsto no programa, de 60 horas. Idealmente, os feedbacks deveriam ser testados em estrutura de tamanho similar à original.

Uma limitação complementar importante de ser exposta é a estrutura do ambiente de testes, o Aprender2. Por maior quantidade de semelhanças que este possui com o ambiente de produção, ou seja, qual os estudantes de fato utilizam em seu rotina de estudos, são diferentes. A maior diferença e que impacta diretamente nos testes é o fato de que, no ambiente de testes, a aba em que o *plugin* situa-se fica recolhida por padrão, havendo a necessidade do estudante acionar um botão no canto superior direito da página para deixá-lo visível, diferente do ambiente de produção em que o *plugin* e os feedbacks ficam, por padrão, habilitados para visualização. Mesmo com essa informação e todas as instruções necessárias fornecidas aos voluntários, uma ação a mais para poder visualizar o AsPIn pôde ter sido impactante nos resultados obtidos,

como foi expressa em forma de comentário por um dos pesquisados.

Capítulo 9

Conclusão

A realização deste trabalho intenta e apresenta uma variedade de níveis de informação elaborativa, incorporada em um único *plugin* que pode ser acessado de acordo com o nível de realização e desempenho do aluno, bem como a natureza da tarefa para o grupo de estudantes da disciplina introdutória à computação. Esses feedbacks obtiveram uma alta taxa de aprovação de acordo com o experimento realizado, o que indica que possui potencial de aceitação e de melhora real no rendimento dos discentes.

À medida que os sistemas adaptativos se tornam mais poderosos e capazes de oferecer um feedback personalizado, baseado em fatores como conhecimento, momento a ser apresentado, sequenciamento, valor de conquista acadêmica, valor ao curso e tamanho, é esperado que os níveis de aprendizado aumentem. Alunos obtendo maiores notas e, conseqüentemente, diminuindo a alta taxa de reprovação que existe atualmente são os objetivos principais deste trabalho. Os resultados mostraram o potencial dos feedbacks informativos e motivacionais, apresentados aos alunos que cursavam a disciplina naquele momento.

Para estudos e desenvolvimentos futuros, as sugestões dos pesquisados possuem grande valia por serem justamente o foco principal da pesquisa. Apresentar de forma sucinta o próximo conteúdo a ser realizado, fornecer ao usuário a opção de desabilitar o *plugin*, caso este ache o irrelevante, coletar mais dados de novos voluntários, fornecer ainda mais exercícios extras, aumentar a visibilidade do *plugin*, acrescentar humor para desempenhos notáveis, são fatores que podem ser acrescentados/aprimorados aos feedbacks. Em adição, com a possibilidade de diferenciá-los e personalizá-los ainda mais futuramente, há a viabilidade de unir os dados coletados de alunos cotistas [31] no formulário de percepção com alunos com maior propensão à enfrentar dificuldades [4], já fornecendo uma estrutura com maior incentivo a estes estudantes, antes mesmo da primeira lista avaliativa. Além disso, os feedbacks ainda possuem a capacidade de serem especificados ainda mais, para o desempenho abaixo da média, em dois diferentes grupos, como é dividido na Universidade de Brasília: II (inferior), para notas de 0,1 a 2,9 e MI (média inferior), para notas de 3 a 4,9. Porém, é necessária uma pesquisa de relevância e custo

benefício de tal diferenciação nos termos pedagógicos.

Referências

- [1] Watson, Christopher e Frederick W.B. Li: *Failure rates in introductory programming revisited*. Em *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ITiCSE '14, página 39–44, New York, NY, USA, 2014. Association for Computing Machinery, ISBN 9781450328333. <https://doi.org/10.1145/2591708.2591749>. 1, 5
- [2] Becker, Brett A. e Keith Quille: *50 years of cs1 at sigcse: A review of the evolution of introductory programming education research*. Em *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, página 338–344, New York, NY, USA, 2019. Association for Computing Machinery, ISBN 9781450358903. <https://doi.org/10.1145/3287324.3287432>. 1, 5
- [3] Silva, Priscylla, Evandro Costa e Joseana Régis de Araújo: *An adaptive approach to provide feedback for students in programming problem solving*. Em Coy, Andre, Yugo Hayashi e Maiga Chang (editores): *Intelligent Tutoring Systems*, páginas 14–23, Cham, 2019. Springer International Publishing, ISBN 978-3-030-22244-4. 1, 5, 6, 8
- [4] Macedo, Fernanda, Lucas Yamin e Maristela de Holanda: *Aspin: Um plugin no moodle para o ensino introdutório de programação para alunos da matemática*. 2021. 3, 5, 10, 26
- [5] Community, Moodle: *Development - moodle docs*, 2022. https://docs.moodle.org/dev/Main_Page. 3
- [6] UnB, FT: *Faculdade de tecnologia cursos de graduação*, 2022. http://www.ft.unb.br/index.php?option=com_content\&view=article&id=5&Itemid=138. 3
- [7] UnB, MAT: *Faculdade de tecnologia cursos de graduação*, 2022. <https://mat.unb.br/graduacao-cursos>. 3
- [8] Cohen, Vicki Blum: *A reexamination of feedback in computer-based instruction: Implications for instructional design*. Educational Technology archive, 25:33–37, 1985. 4, 6
- [9] Black, Paul e Dylan Wiliam: *Assessment and classroom learning*. *Assessment in Education: Principles, Policy & Practice*, 5(1):7–74, 1998. <https://doi.org/10.1080/0969595980050102>. 4

- [10] Shute, Valerie J. e Yoon Jeon Kim: *Formative and Stealth Assessment*, páginas 311–321. 2013. <https://app.dimensions.ai/details/publication/pub.1046591080>. 4, 6
- [11] Phye, Gary D. e Cheryl E. Sanders: *Advice and feedback: Elements of practice for problem solving*. *Contemporary Educational Psychology*, 19(3):286–301, 1994, ISSN 0361-476X. <https://www.sciencedirect.com/science/article/pii/S0361476X84710228>. 4, 8
- [12] Shute, Valerie J.: *Focus on formative feedback*. *Review of Educational Research*, 78(1):153–189, 2008. <https://doi.org/10.3102/0034654307313795>. 4, 6, 11, 12
- [13] Song, Sang H. e John M. Keller: *Effectiveness of motivationally adaptive computer-assisted instruction on the dynamic aspects of motivation*. *Educational Technology Research and Development*, 49, 2001. <https://doi.org/10.1007/BF02504925>. 4, 6
- [14] Hao, Qiang e Michail Tsikerdekis: *How automated feedback is delivered matters: Formative feedback and knowledge transfer*. Em *2019 IEEE Frontiers in Education Conference (FIE)*, página 1–6. IEEE Press, 2019. <https://doi.org/10.1109/FIE43999.2019.9028686>. 4, 6, 8
- [15] Kulhavy, Raymond W.: *Feedback in written instruction*. *Review of Educational Research*, 47(2):211–232, 1977, ISSN 00346543, 19351046. <http://www.jstor.org/stable/1170128>, acesso em 2023-01-30. 4, 6, 12
- [16] Mory, Edna H.: *The use of informational feedback in instruction: Implications for future research*. *Educational Technology Research and Development*, 40(3):5–20, 1992, ISSN 10421629, 15566501. <http://www.jstor.org/stable/30220010>, acesso em 2023-01-30. 4, 6
- [17] Sarsar, Firat: *Student and instructor responses to emotional motivational feedback messages in an online instructional environment*. *Turkish Online Journal of Educational Technology*, 16:115–127, janeiro 2017. 4, 6, 11
- [18] Dolin, Jens, Paul Black, Wynne Harlen e Andrée Tiberghien: *Exploring Relations Between Formative and Summative Assessment*, páginas 53–80. Springer International Publishing, Cham, 2018, ISBN 978-3-319-63248-3. https://doi.org/10.1007/978-3-319-63248-3_3. 4, 5, 6
- [19] Goodman, Jodi, Robert Wood e Margaretha Hendrickx: *Feedback specificity, exploration, and learning*. *The Journal of applied psychology*, 89:248–62, maio 2004. 4
- [20] Meier, Heidi e Marina Lepp: *Effectiveness of feedback based on log file analysis in introductory programming courses*. *Journal of Educational Computing Research*, 0(0):07356331221132651, 0. <https://doi.org/10.1177/07356331221132651>. 4, 5, 6, 9

- [21] Tosi, Henry: *A theory of goal setting and task performance*. Academy of Management Review, 16:480–483, abril 1991. 5
- [22] Law, Kris, Victor Lee e Y.T. Yu: *Learning motivation in e-learning facilitated computer programming courses*. Computers Education, 55:218–228, agosto 2010. 5
- [23] Narciss, Susanne: *Feedback Strategies for Interactive Learning Tasks*, páginas 125–144. janeiro 2008. 5, 6, 11
- [24] Azevedo, Roger e Robert Bernard: *A meta-analysis of the effects of feedback in computer-based instruction*. Journal of Educational Computing Research - JEDUC COMPUT RES, 13:1–1, setembro 1995. 6
- [25] Mcconlogue, Teresa: *Assessment and Feedback in Higher Education: A Guide for Teachers*. maio 2020, ISBN 9781787353657. 6, 23
- [26] Mason, B. e Roger Bruning: *Providing feedback in computer-based instruction: What the research tells us*. Center for Instructional Innovation, 15, janeiro 2001. 6
- [27] Hassan, Lobna, Antonio Dias e Juho Hamari: *How motivational feedback increases user's benefits and continued use: A study on gamification, quantified-self and social networking*. International Journal of Information Management, 46:151–162, junho 2019. 6
- [28] Willemsen, Martijn, Mark Graus e Bart Knijnenburg: *Understanding the role of latent feature diversification on choice difficulty and satisfaction*. User Modeling and User-Adapted Interaction, 26, outubro 2016. 6, 12
- [29] Miller, Tess: *Formative computer-based assessment in higher education: The effectiveness of feedback in supporting student learning*. Assessment Evaluation in Higher Education - ASSESS EVAL HIGH EDUC, 34:181–192, abril 2009. 8
- [30] Joshi, Ankur, Saket Kale, Satish Chandel e Dinesh Pal: *Likert scale: Explored and explained*. British Journal of Applied Science Technology, 7:396–403, janeiro 2015. 17
- [31] Holanda, Maristela, George von Borries, Dilma Silva, Camilo Dorea, Roberta Oliveira e Edison Ishikawa: *The intellectual sense of belonging and self-efficacy in the introduction to computer science courses at university of brasilia in brazil*. páginas 1–8, outubro 2020. 26