Tradução e validação de um inventário de conceitos sobre programação introdutória

Ana Caroline R. Braz[1], Leandro S. G. Carvalho[1], Elaine H. T. Oliveira[1], David B. F. Oliveira[1], Roberto A. Bittencourt[2], Bianca L. Santana[2], Filipe Dwan Pereira[3]

[1]Universidade Federal do Amazonas (UFAM)
Av. Gen. Rodrigo Octávio 6200 – Coroado I – CEP 69.080-900 – Manaus – AM – Brasil

[2]Universidade Estadual de Feira de Santana (UEFS)
Av. Transnordestina – Novo Horizonte – CEP 44036-900 – Feira de Santana – BA – Brasil

[3]Universidade Federal de Roraima (UFRR)
Av. Cap. Ene Garcês 2413 – Aeroporto – CEP 69310-000 – Boa Vista – RR – Brasil

{ana.braz,galvao,elaine,david}@icomp.ufam.edu.br

roberto@uefs.br, biancasantana.ls@gmail.com, filipe.dwan@ufrr.br

Resumo. *Um inventário de conceitos (IC) é um conjunto de questões de múltipla escolha ou discursivas com intuito de medir o conhecimento dos alunos em determinado tópico ou assunto. Visto que, no português brasileiro, não existe nenhum IC criado no tópico de programação introdutória, o objetivo principal desse trabalho visa traduzir e validar um inventário de conceitos independente de linguagem de programação. Como resultado, obtivemos um Alfa de Cronbach elevado, porém baixas correlações com as notas dos alunos.*

Abstract. *A concept inventories (CI) is a set of multiple-choice or discursive questions intended to measure students' knowledge of a particular subject or subject. Since, in Brazilian Portuguese, there is no CI created in the introductory programming topic, the main objective of this work is to translate and validate an inventory of independent programming language concepts. As a result, we obtained a high Cronbach's Alpha, but low correlations with student grades.*

1. Introdução

Conhecida na literatura como CS1, a disciplina de Introdução à Computação possui grande complexidade para os alunos de graduação [Pereira et al. 2019, Lima et al. 2020, Araujo et al. 2021], pois demanda alta capacidade cognitiva para compreensão dos problemas e escrita dos códigos necessários para a resolução [Fonseca et al. 2019, Robins 2019, Luxton-Reilly et al. 2018, Lima et al. 2020, Lima Lima et al. 2021, Pereira et al. 2020b, Freitas Júnior et al. 2020]. Isso vale tanto para a área de computação, quanto para outros campos das ciências exatas e engenharias nos quais a disciplina CS1 é obrigatória [Pereira et al. 2020a, Santos et al. 2020, Araujo et al. 2021, Costa et al. 2021]. No caso desses últimos, que são conhecidos por *non-CS-majors* por não terem a computação como atividade fim, a taxa de reprovação em CS1 pode ser superior a 50% [Alves et al. 2019, Fonseca et al. 2020, Pereira et al. 2021].

DOI: 10.5753/sbie.2021.218453

De fato, percebe-se que muitos dos alunos, ao final da disciplina, podem possuir pouco entendimento concreto sobre os conceitos, levando-os a ter concepções errôneas [McCracken et al. 2001, Lister et al. 2004, Robins 2019, Caceffo et al. 2019, Braz et al. 2021]. Sendo assim, a criação de um inventário de conceitos (IC) para programação introdutória passa a ser fundamental. Os ICs, além de permitirem que tais concepções errôneas sejam conhecidas e metodologias e intervenções sejam aplicadas para que estas possam ser corrigidas, são essenciais para a comparação do ensino entre turmas distintas de CS1 e avaliar o conhecimento do aluno em determinado tópico ou assunto [Mühling et al. 2015, Parker et al. 2016, Caceffo et al. 2016, Tew 2010], sendo essas também razões válidas como motivação para este artigo.

Mais especificamente, no tópico de programação introdutória, não há instrumento validado para o português do Brasil e de uso aberto. Visto isso, o objetivo principal do trabalho é validar a tradução, para o português brasileiro, do inventário de conceitos de programação proposto por [Parker et al. 2016]. Foi escolhido esse questionário por ser uma replicação do primeiro IC criado independente de linguagem para programação introdutória. Com intuito de manter a isomorfia dos instrumentos, será utilizada a mesma metodologia proposta por [Parker et al. 2016].

O artigo está organizado da seguinte forma. Na Seção 2, serão apresentados conceitos e terminologias que serão utilizadas no decorrer do texto. Na Seção 3, serão apresentados os trabalhos relacionados. Na Seção 4, será apresentada a metodologia. Na Seção 5, será apresentado os resultados da pesquisa. Na Seção 6, serão apresentados limitações e ameaças à validade. E, por fim, na Seção 7, a conclusão e trabalhos futuros.

2. Conceitos e Terminologias

Nesta seção, será descritas conceitos e terminologias utilizadas no decorrer do artigo.

2.1. Inventário de Conceitos

Um inventário de conceitos (IC) é um conjunto de questões de múltipla escolha ou discursivas que possuem o objetivo de identificar concepções errôneas (em inglês, *misconceptions*) ou medir o conhecimento dos alunos em determinado tópico ou assunto [Caceffo et al. 2016, Parker et al. 2016, Tew 2010].

No cenário de computação, o primeiro inventário de conceitos criado foi o *Foundational CS1 Assessment* (FCS1) de Alisson Tew [Parker et al. 2016, Tew et al. 2005, Tew and Guzdial 2011, Tew 2010] e, logo após, esse trabalho foi adaptado por Miranda Parker, criando o *Second CS1 Assessment*(SCS1). Neste artigo, o inventário de [Parker et al. 2016] foi usado como base para a tradução e validação.

2.2. Método *Think-Aloud*

Think-Aloud é um método para estudar processos mentais em que os participantes são solicitados a fazer comentários em voz alta enquanto trabalham em uma tarefa [Lewis 1982]. Durante esse processo, uma outra pessoa fica observando e perguntando sobre o que o participante está tentando fazer e seus pensamentos. Com esse método, é possível identificar erros de digitação, palavras inadequadas e questões que não possuem um nível de dificuldade apropriado [Parker et al. 2016, Lewis 1982].

Neste artigo, o método foi realizado junto aos alunos já aprovados na matéria de CS1, a fim de encontrar erros e mal-entendidos, para que pudessem ser corrigidos para aplicar aos alunos iniciantes da matéria.

2.3. Teoria de Resposta ao Item (TRI)

A Teoria de Resposta ao Item é utilizada de forma quantitativa, ou seja, ela mede a dificuldade e a discriminação de uma questão. Sendo assim, para mensurar a dificuldade usa-se a porcentagem de participantes que conseguiram responder corretamente a questão. A discriminação mede o desempenho do aluno em uma determinada questão e sua predição para o desempenho no questionário em geral. O ideal para uma questionário de múltipla-escolha contendo 5 opções é de dificuldade entre 70-71% e discriminação "bom" [Parker et al. 2016, Mühling et al. 2015, Hambleton et al. 1991, Lord 1952].

O TRI, no artigo, foi utilizado durante a fase de aplicação nas turmas de testes do questionário e durante a fase final do trabalho para análise dos resultados.

2.4. Alfa de Cronbach

O Alfa de Cronbach é uma medida de confiabilidade usada para medir a consistência de um questionário. Essa medida é dada pela seguinte fórmula:

$$\alpha = \frac{N^2\overline{Cov}}{\sum s^2_{Item} + \sum Cov_{Item}} \tag{1}$$

Observa-se que na fórmula é possível calcular a variância dentro de um item ($\sum s^2_{Item}$) e a covariância entre um item particular e qualquer outro item ($\sum Cov_{Item}$). Sendo assim, o numerador é dado pelo quadrado do número de questões multiplicado pela média das covariâncias entre os itens. O denominador é dado pela soma das variâncias e covariâncias do item [Field 2013].

No geral, uma medida aceitável para o Alfa de Cronbach fica entre 0,7 - 0,8, sendo que valores abaixo disso são considerados não confiáveis. Para testes que envolvem habilidades, o ponto de corte mais adequado é de 0,7 [Field 2013]. O Alfa de Cronbach foi utilizado para poder verificar a confiabilidade do material traduzido e replicado do experimento de [Parker et al. 2016].

3. Trabalhos Relacionados

Em 1992, o primeiro inventário de conceitos foi criado no campo da Física, com o objetivo de ajudar os professores a sondar e avaliar as concepções errôneas de seus alunos [Hestenes et al. 1992]. Desde então, várias outras áreas tentaram replicar esse trabalho em diversos tópicos, principalmente naqueles que são considerados fundamentais e complexos de serem entendidos. O objetivo principal dos trabalhos de replicação, como por exemplo de [Parker et al. 2016] na área de computação, era medir o conhecimento dos alunos em determinado tópico ou assunto.

Em 2010, Allison Tew criou o primeiro IC independente de linguagem de programação e validado para CS1, conhecido como *Foundational CS1 Assessment* (FCS1) [Parker et al. 2016, Tew et al. 2005, Tew and Guzdial 2011, Tew 2010]. O questionário,

que é baseado em pseudocódigo, possui um total de 27 questões que envolvem fundamentos, operadores lógicos, *loops* definidos e indefinidos, *arrays*, funções e recursão. Em sua tese, Tew mostra que 24 das 27 questões possuem uma discriminação forte, nível adequado de dificuldade e baixa probabilidade de "adivinhação". Com um total de 931 participantes, foi realizada a correlação de Pearson entre a nota do questionário com as notas do curso introdutório de programação. Chegou-se a uma correlação positiva, *Pearson's r*(931) = 0,499, $p \leq 0{,}001$ [Tew 2010, Tew and Guzdial 2011].

Em 2016, Miranda Parker e colegas replicaram o trabalho de Tew, criando o *Second CS1 Assessment* (SCS1) [Parker et al. 2016]. Em vista do FSC1 não ser aberto publicamente, a fim de evitar que o questionário ou as respostas sejam facilmente encontrados diminuindo a sua eficácia, Parker e colegas criaram uma versão isomórfica do FCS1. Nessa versão, algumas palavras do problema, variáveis e opções de respostas são alteradas, no entanto é mantida a área do assunto e o estilo das perguntas. O questionário, que contém questões em pseudocódigo, tem o mesmo objetivo do FSC1: medir o conhecimento sobre os conceitos de programação introdutória no nível de graduação de maneira independente de linguagem [Parker et al. 2016, Tew 2010, Tew and Guzdial 2011]. A metodologia utilizada por eles foi:

1. Criação de uma versão isomórfica do FSC1: Nova versão chamada de *Second CS1 Assessment* (SCS1).
2. Entrevistas com 3 alunos de graduação matriculados em aulas introdutórias de computação pelo método *Think-Aloud* [Lewis 1982]: Durante as entrevistas, foi possível verificar que o questionário SCS1 possuía alguns erros de digitação e algumas palavras vagas.
3. Estudo de validação: Foi aplicado os questionários FSC1 e SCS1 em um grupo de estudantes dos cursos de computação e engenharias (n = 183). Os alunos, divididos em 2 turmas, realizaram os questionários de maneira alternada. Na primeira semana, a turma 1 respondeu o questionário FSC1 e a turma 2, o questionário SCS1. Na semana seguinte, houve a troca.
4. Correlação com o FSC1: Utilizando a correlação de Pearson foi possível verificar uma forte correlação positiva *Pearson's r*(183) = 0,566, $p = 0{,}000$.
5. Análise quantitativa usando a Teoria de Resposta ao Item (TRI): Apesar da correlação positiva, ambos questionários tiveram níveis de dificuldade "difícil" e a maioria das questões teve sua discriminação "razoável", definidos pela Teoria de Resposta ao Item [Field 2013].
6. Alfa de Cronbach: Usado como uma medida de confiabilidade com base em correlações entre diferentes itens de uma avaliação, o Alfa de Cronbach do FCS1 foi de 0,53 e do SCS1 de 0,59.

Consequentemente ambos os resultados são considerados abaixo do nível de aceitação (0,65 [Field 2013]), o que sugere um refinamento em ambos questionários e re-teste usando uma amostra maior de alunos. Diante disso, neste estudo nós replicamos o questionário SCS1, a fim de verificar se os resultados apresentados em um contexto educacional presencial, também se aplicam ao nosso contexto educacional a distância, em vista da pandemia.

4. Metodologia

A seguir será apresentada a metodologia utilizada neste estudo, que por fins de isomorfia dos instrumentos, segue as mesmas etapas expressas no estudo de [Parker et al. 2016].

4.1. Tradução do Questionário para o Português Brasileiro

Primeiramente, entrou-se em contato com a primeira autora do trabalho [Parker et al. 2016], visto que o questionário não é aberto publicamente. Ao recebê-lo, foi feita uma tradução minuciosa do questionário para o português brasileiro, o qual será chamado de SCS1-pt no decorrer do artigo.

A tradução foi revisada por 3 pessoas, fluentes em inglês e português. A primeira pessoa realizou a tradução do questionário, que logo após foi aplicado em uma turma teste. Notou-se que uma das opções de resposta de uma das questões estava traduzida errada, com isso uma segunda pessoa revisou minuciosamente. Com efeito, pequenas falhas (e.g. ponto ao invés vírgula) foram corrigidas a fim de aprimorar a compreensão das questões. Após a última revisão pela terceira pessoa, o questionário foi preparado para a aplicação em alunos aprovados em CS1 por meio do método *Think-Aloud*.

4.2. Aplicação em alunos aprovados em CS1 por meio do método *Think-Aloud*

Foi enviado um e-mail para todos os alunos da graduação de uma universidade pública federal à procura de 10 voluntários, que já foram aprovados na matéria de CS1, para realização do questionário. Como forma de retribuição pela participação foram disponibilizados certificados de horas-aulas de 4 horas para os participantes.

Em vista da pandemia, todas as entrevistas foram realizadas via videoconferência de forma individual. O questionário, feito pelo Google Form, consistia em 6 perguntas que estavam contidas no SCS1-pt. Com esse número, foi possível revisar todas as questões, pelo menos, 2 vezes. No total, 9 questionários foram criados e as questões separadas da seguinte forma:

- Questionário 1: 1, 2, 3, 4, 5, 6;
- Questionário 2: 4, 5, 6, 7, 8, 9;
- Questionário 3: 7, 8, 9, 10, 11, 12;
- Questionário 4: 10, 11, 12, 13, 14, 15;
- Questionário 5: 13, 14, 15, 16, 17, 18;
- Questionário 6: 16, 17, 18, 19, 20, 21;
- Questionário 7: 19, 20, 21, 22, 23, 24;
- Questionário 8: 22, 23, 24, 25, 26, 27;
- Questionário 9: 25, 26, 27, 1, 2, 3.

Cada questionário foi respondido por 1 aluno, totalizando 10 entrevistas. O 10º aluno respondeu o questionário que houve mais problemas, no caso, o questionário 2. Isso ocorreu, pois o voluntário que respondeu esse questionário, apesar de aprovado na matéria de CS1, não possuía o domínio dos assuntos tratados nas questões. 7 dos 10 voluntários obtiveram acertos acima de 4. Ao final do questionário, os alunos responderam um pequeno questionário sociodemográfico. Dentre os 10 voluntários, 5 possuíam mais afinidade com a linguagem de programação C, 4 com Python e 1 com JavaScript.

Durante a aplicação, os alunos encontraram alguns mal-entendidos e erros de digitação pelos alunos durante a aplicação. Sendo assim, todas as questões foram revisadas novamente, a fim de corrigir todos os erros, para assim prosseguir para uma aplicação em uma turma teste.

4.3. Aplicação em uma turma teste

Após a revisão, o SCS1-pt foi implementado em HTML e CSS, para ser adicionado a uma plataforma de juiz online e aplicado em uma turma teste.

Durante o período 2020/2, a Universidade utilizou a modalidade de Ensino Remoto Emergencial (ERE). Nesse período, 2 turmas foram escolhidas para aplicação do questionário. Como forma de motivação, optou-se por tornar o instrumento avaliativo, isto é, ao responder o questionário o aluno recebia 1,0 ponto de bônus (0-10) na média final. Um total de 38 alunos participaram do questionário.

O questionário de 27 questões teve duração de 1 hora para ser respondido. Ao final, foi adicionado um campo para os alunos compartilharem opiniões, as principais dificuldades e sugestões para o melhoramento das questões a respeito do questionário. 37% dos alunos relataram ter muita dificuldade ao responder o questionário por conta ou da linguagem ou do tamanho das questões junto ao tempo curto. Utilizando a Teoria de Resposta ao Item (TRI), foi possível observar que o questionário está no nível de discriminação "bom" e de dificuldade "difícil" (Tabela 1).

Tabela 1. Classificações da TRI das questões SCS1-pt.

		Dificuldade (0 – 100%)			
		Difícil 0 – 50%	Moderado 50 – 85%	Fácil 85 – 100%	Total
Discriminação	Ruim (<0,1)	25	—	—	1 item
	Razoável (0,1 – 0,3)	19, 20, 27	—	—	3 itens
	Bom (>0,3)	3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 21, 22, 23, 24, 26	1, 2	—	23 itens
	Total	25 itens	2 itens	0	27 itens

4.4. Aplicação da avaliação aos alunos iniciantes da disciplina de CS1

Após a nova revisão, questões envolvendo funções e recursões foram retiradas do questionário, pois não fazem parte da ementa de CS1 para *non-majors*. Logo, um subconjunto de 18 questões do SCS1-pt foi testado. Os tipos de questões permaneceram com a mesma distribuição, sendo 6 questão de definição (*definitional*), 6 de rastreamento (*tracing*) e 6 de compilação do código (*code compilation*).

Para aplicação da avaliação, no início do semestre, foi utilizada uma plataforma de juiz online. Desta vez o tempo para execução do questionário foi de 2 horas, que anteriormente era de 1 hora, a fim de que houvesse tempo para leitura dos textos e códigos ou caso algum aluno tivesse problema de conexão de internet não fosse prejudicado.

4.5. Análise dos Resultados com TRI e Alfa de Cronbach

Para analisar os resultados finais, foi usada a Teoria de Resposta ao Item para medir a discriminação e dificuldade das questões, e o Alfa de Cronbach para a confiabilidade do questionário. Ambos são recomendadas pela literatura educacional para este fim [Mühling et al. 2015].

De acordo com [Lord 1952, Parker et al. 2016] um questionário ideal deve ter um nível de dificuldade entre 70-74% e um nível de discriminação "bom". E segundo [Cortina 1993, Field 2013], questionários com mais de 12 itens e correlações altas podem ser fatores para um Alfa de Cronbach por volta de 0,7 (0,65 a 0,84).

5. Resultados

Durante o período 2020/1, realizado no início de 2021, 7 turmas de CS1 foram ofertadas para *non-majors*. No decorrer do semestre, 8 listas foram realizadas gerando 8 notas e uma média entre elas. Nesta etapa, um total de 185 alunos responderam o questionário.

Para a análise das correlações, foi utilizada a linguagem *Python* e as bibliotecas *Pandas*, para leitura de arquivos em .csv e manipulação de dados, e *Pingouin*, para o cálculo das correlações. Visto que todas as distribuições das notas não obedeciam uma distribuição normal, foi utilizado a correlação de Spearman.

Infelizmente, as correlações encontradas foram fracas ou não significativas [Akoglu 2018] (Tabela 2). De fato, 50% dos respondentes, ao final do questionário, relataram a falta de conhecimento sobre o assunto e grande dificuldade em entender a linguagem em pseudocódigo. Alguns dizem até ter tentado entender a lógica, porém por conta do tempo tiveram que "adivinhar" as respostas.

Tabela 2. Valor-p de Spearman e Correlações

		Valor p	Correlação
Notas dos trabalhos	nota1	0,450	0,056
	nota2	0,060	0,138
	nota3	0,029	0,161
	nota4	0,286	0,079
	nota5	0,109	0,118
	nota6	0,030	0,160
	nota7	0,065	0,136
	nota8	0,033	0,157
	média	0,022	0,168

Utilizando a Teoria de Resposta ao Item, foi possível perceber que o questionário teve um nível de discriminação "bom" e dificuldade "difícil" (Tabela 3), mantendo o nível quando aplicado na turma teste.

Utilizando o Alfa de Cronbach foi possível encontrar um valor de 0,932, mostrando uma alta confiabilidade [Field 2013]. Visto que o α depende da quantidade de itens, é válido dizer que, quanto maior o número de itens, maior será o Alfa de Cronbach. Portanto, pode-se dizer que este número se dá por causa da quantidade de questões presentes no questionário.

Tabela 3. Classificações da TRI das questões SCS1-pt.

		Dificuldade (0 – 100%)			
		Difícil 0-50%	Moderado 50-85%	Fácil 85-100%	Total
Discriminação	Ruim (<0,1)	—	—	—	0 item
	Razoável (0,1 – 0,3)	4	—	—	1 item
	Bom (>0,3)	1, 2, 3, 5, 6, 7, 8, 9, 10 11, 12, 13, 14, 15, 16, 17, 18	—	—	17 itens
	Total	18 itens	0 item	0 item	18 itens

Com esses resultados, pode-se observar que o questionário não funciona como o esperado em um cenário educacional de ensino remoto com alunos de cursos *non-CS*. Em vista da pandemia, o cenário se torna mais desafiador, além dos fatores externos que ela nos trouxe, os alunos de cursos *non-CS* podem não ter a motivação intrínseca de aprender a programar. Já [Parker et al. 2016] conseguiram trabalhar em um cenário educacional presencial, o que pode ter sido um fator positivo para as correlações positivas.

6. Limitações e Ameaças à validade

O Alfa de Cronbach também pode ter uma segunda interpretação que mede a "unidimensionalidade" [Field 2013]. Isso acontece quando existe um fator subjacente aos dados, logo o α será uma medida da força daquele fator [Field 2013, Cortina 1993]. Apesar disso, é possível obter um Alfa de 0,8 com dois fatores não-correlacionados [Field 2013]. Além de que questionários com mais de 12 itens e correlações altas também podem ser fatores para o Alfa alcançar valores por volta de 0,7 (0,65 a 0,84) [Cortina 1993]. Isso mostra que não se deve usá-lo como medida de "unidimensionalidade". Sugere-se que se existem vários fatores, a fórmula deve ser aplicada separadamente a itens relacionados a diferentes fatores [Field 2013].

Ademais, durante toda pesquisa, foi utilizada uma amostragem por conveniência. Sendo assim, os dados finais podem ser tendenciosos [Guimarães 2008]. Visto que o instrumento foi aplicado de forma avaliativa, isto é, ao responder o questionário, o aluno recebia 1,0 ponto de bônus (0-10) na média final, pode-se dizer que alguns alunos não responderam o questionário com muito afinco. Foi o caso de 24 alunos que apenas pularam as questões sem ao menos ler, gerando nota zero.

7. Conclusão e Trabalhos futuros

Nesse artigo foi apresentada a tradução e validação do inventário de conceitos SCS1, de Miranda Parker, para o português brasileiro, utilizando da mesma metodologia e mantendo a isomorfia dos instrumentos.

Os resultados mostram que, apesar da correlação fraca, os níveis de discriminação e de dificuldade seguem sendo os mesmos aplicados durante a fase da turma teste, "bom"

e "difícil". E o Alfa de Cronbach permaneceu alto, mostrando uma alta confiabilidade no material traduzido. Observa-se que em um cenário educacional de ensino remoto, o questionário não funciona tão bem, visto que alunos de cursos *non-CS* podem não ter a motivação intrínseca de aprender a programar.

Para futuros trabalhos, pode-se levar em consideração uma quantidade maior de alunos, com modelo de ensino presencial, com o objetivo de avaliar o ganho de aprendizagem ou para uma validação que possua uma correlação maior com as notas do semestre. Dessa forma, poderemos verificar se os resultados apontados em [Parker et al. 2016] se manterão ou se as correlações fracas serão encontradas como as reportadas neste estudo.

Agradecimentos

Os autores agradecem ao apoio e financiamento prestado pela Universidade Federal do Amazonas – UFAM por meio do Edital 081/2019 – PROPESP/UFAM, do Programa Institucional de Bolsas de Iniciação Científica (PIBIC). Além disso, esta pesquisa, realizada no âmbito do Projeto Samsung-UFAM de Ensino e Pesquisa (SUPER), de acordo com o artigo 48 do Decreto no 6.008/2006 (SUFRAMA), foi parcialmente financiada pela Samsung Electronics da Amazônia Ltda., nos termos de Lei Federal no 8.387/1991, mediante contrato 001/2020, firmado com a Universidade Federal do Amazonas e a FAEPI, Brasil. Contamos também com o apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001 e do Conselho Nacional de Desenvolvimento Científico e Tecnológico - Brasil (Processo 308513/2020-7).

Referências

Akoglu, H. (2018). User's guide to correlation coefficients. *Turkish Journal of Emergency Medicine*, 18(3):91–93.

Alves, A., Carvalho, L. S. G., Oliveira, E., and Fernandes, D. (2019). Análise comportamental em juízes online para predição do desempenho final de alunos em disciplinas de computação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 1906.

Araujo, A., Zordan Filho, D. L., Oliveira, E. H. T., Carvalho, L. S. G., Pereira, F. D., and Oliveira, D. B. F. (2021). Mapeamento e análise empírica de misconceptions comuns em avaliações de introdução à programação. In *Anais do Simpósio Brasileiro de Educação em Computação*, pages 123–131. SBC.

Braz, A. C., Carvalho, L., Oliveira, E., Oliveira, D., Pereira, F., Bittencourt, R., and Santana, B. (2021). Validação e analise de um inventario de conceitos sobre programação introdutoria. In *Anais Estendidos do Simpósio Brasileiro de Educação em Computação*, pages 27–28, Porto Alegre, RS, Brasil. SBC.

Caceffo, R., Frank-Bolton, P., Souza, R., and Azevedo, R. (2019). Identifying and validating java misconceptions toward a cs1 concept inventory. In *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, ITiCSE '19, page 23–29, New York, NY, USA. Association for Computing Machinery.

Caceffo, R., Wolfman, S., Booth, K. S., and Azevedo, R. (2016). Developing a computer science concept inventory for introductory programming. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, pages 364–369.

Cortina, J. M. (1993). What is coefficient alpha? an examination of theory and applications. *Journal of applied psychology*, 78(1):98.

Costa, T. L., de Oliveira, E. H. T., Passito, A., de Souza Pinto, M. A., de Carvalho, L. S. G., de Oliveira, D. B. F., and Pereira, F. D. (2021). Material didático interativo para a disciplina de introdução à programação de computadores. In *Anais Estendidos do Simpósio Brasileiro de Educação em Computação*, pages 41–42. SBC.

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. sage.

Fonseca, S., Oliveira, E., Pereira, F., Fernandes, D., and Carvalho, L. S. G. (2019). Adaptação de um método preditivo para inferir o desempenho de alunos de programação. In *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, volume 30, page 1651.

Fonseca, S. C., Pereira, F. D., Oliveira, E. H., Oliveira, D. B., Carvalho, L. S., and Cristea, A. I. (2020). Automatic subject-based contextualisation of programming assignment lists. EDM.

Freitas Júnior, H. B., Pereira, F. D., Oliveira, E. H. T., Oliveira, D. B. F., and Carvalho, L. S. G. (2020). Recomendação automática de problemas em juízes online usando processamento de linguagem natural e análise dirigida aos dados. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1152–1161. SBC.

Guimarães, P. R. B. (2008). Métodos quantitativos estatísticos.

Hambleton, R. K., Shavelson, R. J., Webb, N. M., Swaminathan, H., and Rogers, H. J. (1991). *Fundamentals of item response theory*, volume 2. Sage.

Hestenes, D., Wells, M., and Swackhamer, G. (1992). Force concept inventory. *The physics teacher*, 30(3):141–158.

Lewis, C. (1982). *Using the"thinking-aloud"method in cognitive interface design*. IBM TJ Watson Research Center Yorktown Heights, NY.

Lima, M., Carvalho, L. S. G., de Oliveira, E. H. T., Oliveira, D. B. F., and Pereira, F. D. (2020). Classificação de dificuldade de questões de programação com base em métricas de código. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1323–1332. SBC.

Lima Lima, M. A. P., Carvalho, L. S. G., de Oliveira, E. H. T., Oliveira, D. B. F., and Pereira, F. D. (2021). Uso de atributos de código para classificação da facilidade de questões de codificação. In *Anais do Simpósio Brasileiro de Educação em Computação*, pages 113–122. SBC.

Lister, R., Adams, E. S., Fitzgerald, S., Fone, W., Hamer, J., Lindholm, M., McCartney, R., Moström, J. E., Sanders, K., and Seppälä, O. (2004). A multi-national study of reading and tracing skills in novice programmers. *ACM SIGCSE Bulletin*, 36(4):119–150.

Lord, F. M. (1952). The relation of the reliability of multiple-choice tests to the distribution of item difficulties. *Psychometrika*, 17(2):181–194.

Luxton-Reilly, A., Albluwi, I., Becker, B. A., Giannakos, M., Kumar, A. N., Ott, L., Paterson, J., Scott, M. J., Sheard, J., and Szabo, C. (2018). Introductory programming:

a systematic literature review. In *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pages 55–106.

McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y. B.-D., Laxer, C., Thomas, L., Utting, I., and Wilusz, T. (2001). A multi-national, multi-institutional study of assessment of programming skills of first-year cs students. In *Working group reports from ITiCSE on Innovation and technology in computer science education*, pages 125–180.

Mühling, A., Ruf, A., and Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. In *Proceedings of the workshop in primary and secondary computing education*, pages 2–10.

Parker, M. C., Guzdial, M., and Engleman, S. (2016). Replication, validation, and use of a language independent cs1 knowledge assessment. In *Proceedings of the 2016 ACM conference on international computing education research*, pages 93–101.

Pereira, F. D., Fonseca, S. C., Oliveira, E. H., Cristea, A. I., Bellhäuser, H., Rodrigues, L., Oliveira, D. B., Isotani, S., and Carvalho, L. S. (2021). Explaining individual and collective programming students' behaviour by interpreting a black-box predictive model. *IEEE Access*.

Pereira, F. D., Oliveira, E., Cristea, A., Fernandes, D., Silva, L., Aguiar, G., Alamri, A., and Alshehri, M. (2019). Early dropout prediction for programming courses supported by online judges. In *International Conference on Artificial Intelligence in Education*, pages 67–72. Springer.

Pereira, F. D., Oliveira, E. H., Oliveira, D. B., Cristea, A. I., Carvalho, L. S., Fonseca, S. C., Toda, A., and Isotani, S. (2020a). Using learning analytics in the amazonas: understanding students' behaviour in introductory programming. *British Journal of Educational Technology*.

Pereira, F. D., Souza, L. M., Oliveira, E. H. T., Oliveira, D. B. F., and Carvalho, L. S. G. (2020b). Predição de desempenho em ambientes computacionais para turmas de programação: um mapeamento sistemático da literatura. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1673–1682. SBC.

Robins, A. V. (2019). Novice programmers and introductory programming. In *The Cambridge Handbook of Computing Education Research*, chapter 12, pages 327–376. Cambridge University Press, Cambridge.

Santos, I. L., Oliveira, D. B. F., Carvalho, L. S. G., Pereira, F. D., and Oliveira, E. H. T. (2020). Tempos de transição em estados de corretude e erro como indicadores de desempenho em juízes online. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1283–1292. SBC.

Tew, A. E. (2010). *Assessing fundamental introductory computing concept knowledge in a language independent manner*. PhD thesis, Georgia Institute of Technology.

Tew, A. E. and Guzdial, M. (2011). The fcs1: a language independent assessment of cs1 knowledge. In *Proceedings of the 42nd ACM technical symposium on Computer science education*, pages 111–116.

Tew, A. E., McCracken, W. M., and Guzdial, M. (2005). Impact of alternative introductory courses on programming concept understanding. In *Proceedings of the first international workshop on Computing education research*, pages 25–35.