



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS VII - PATOS
CENTRO DE CIÊNCIAS EXATAS E SOCIAIS APLICADAS
CURSO DE GRADUAÇÃO EM BACHARELADO EM COMPUTAÇÃO**

VINICIUS AUGUSTUS ALVES GOMES

**INVESTIGANDO A RELAÇÃO ENTRE UM CONJUNTO DE HABILIDADES
PREDITORAS DE PROGRAMAÇÃO E A ESCRITA/LEITURA DE CÓDIGOS EM
INICIANTES**

**PATOS - PB
2022**

VINICIUS AUGUSTUS ALVES GOMES

**INVESTIGANDO A RELAÇÃO ENTRE UM CONJUNTO DE HABILIDADES
PREDITORAS DE PROGRAMAÇÃO E A ESCRITA/LEITURA DE CÓDIGOS EM
INICIANTES**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Computação do Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

Orientador: MSc. Jucelio Soares dos Santos

**PATOS - PB
2022**

É expressamente proibido a comercialização deste documento, tanto na forma impressa como eletrônica. Sua reprodução total ou parcial é permitida exclusivamente para fins acadêmicos e científicos, desde que na reprodução figure a identificação do autor, título, instituição e ano do trabalho.

G633i Gomes, Vinicius Augustus Alves.

Investigando a relação entre um conjunto de habilidades preditoras de programação e a escrita/leitura de códigos em iniciantes [manuscrito] / Vinicius Augustus Alves Gomes. - 2022.

47 p. : il. colorido.

Digitado.

Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências Exatas e Sociais Aplicadas, 2022.

"Orientação : Prof. Me. Jucelio Soares dos Santos, Coordenação do Curso de Computação - CCEA."

1. Programação. 2. Habilidades preditoras. 3. Correlação.
4. Leitura de Códigos. I. Título

21. ed. CDD 005.1

VINICIUS AUGUSTUS ALVES GOMES

INVESTIGANDO A RELAÇÃO ENTRE UM CONJUNTO DE HABILIDADES
PREDITORAS DE PROGRAMAÇÃO E A ESCRITA/LEITURA DE CÓDIGOS EM
INICIANTES

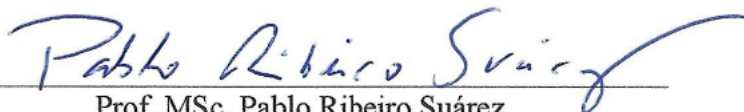
Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Computação do Centro de Ciências Exatas e Sociais Aplicadas da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

Trabalho aprovado em 01/08/2022.

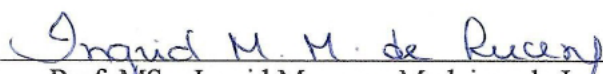
BANCA EXAMINADORA


Prof. MSc. Jucelio Soares dos Santos

(Orientador)


Prof. MSc. Pablo Ribeiro Suárez

(Examinador)


Prof. MSc. Ingrid Morgane Medeiros de Lucena

(Examinadora)

Dedico este trabalho a meus pais, irmãs, amigos e professores. Sem eles ao meu lado, este trabalho não seria possível.

AGRADECIMENTOS

A meus pais, que nunca mediram esforços para assegurar minha educação como prioridade, por proporcionarem a melhor vida que podiam me dar e por incentivarem meus sonhos.

Aos meus amigos, que me acompanham, me incentivam diante das dificuldades e entenderam minha ausência enquanto produzo este trabalho.

Aos meus professores, pelos seus ensinamentos e tempo disponibilizado a mim, permitindo minha evolução como estudante e profissional.

*“Nós só podemos ver um pouco do futuro,
mas o suficiente para perceber que há muito a fazer.”*

Alan Turing

RESUMO

Este trabalho teve como objetivo investigar a relação entre um conjunto de habilidades preditoras de programação e a escrita/leitura de códigos em iniciantes em programação. Para tanto: i) aplicou-se o Screening Programming (um sistema web com validade psicométrica) para estimar habilidades de programação preditiva (resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva); ii) desenvolveu/aplicou-se o "Instrumento de Aptidão em Programação" para estimar as habilidades de escrita/leitura de código; os itens presentes no "Instrumento de Aptidão em Programação" apresentam boa análise de conteúdo e boas propriedades psicométricas, influenciando em sua boa confiabilidade para ser utilizado em avaliações profissionais; E, por fim, iii) com base nas estimativas dos instrumentos, correlacionou-se cada habilidade preditora de programação com as habilidades de escrita/leitura de código dos sujeitos. Como resultado, descobriu-se que existe uma forte correlação entre resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva com a habilidade dos sujeitos em escrever/ler códigos.

Palavras-chaves: Correlação. Habilidades Preditoras de Programação. Escrita/Leitura de Códigos.

ABSTRACT

This work aimed to investigate the relationship between a set of predictive programming skills and code writing/reading in programming beginners. To do so: i) we applied Screening Programming (a web system with psychometric validity) to estimate predictive programming skills (problem-solving, abstract thinking, mathematical reasoning, and cognitive flexibility); ii) we developed/applied the "Instrumento de Aptidão em Programação" to estimate code writing/reading skills; the items present in the "Instrumento de Aptidão em Programação" present good content analysis and good psychometric properties, influencing their good reliability to be used in professional assessments; And, finally, iii) based on the instrument estimates, we correlated each predictive programming skill with the subjects' code writing/reading skills. As a result, we found a strong correlation between problem-solving, abstract thinking, mathematical reasoning, and cognitive flexibility with subjects' ability to write/read code.

Keywords: Correlation. Predictive Programming Skills. Writing/Reading Codes.

LISTA DE ILUSTRAÇÕES

Figura 1 – Página inicial do Screening Programming.	24
Figura 2 – Página de créditos do Screening Programming.	24
Figura 3 – Página inicial do professor com informações sobre os grupos criados.	24
Figura 4 – Página inicial do grupo.	25
Figura 5 – Página de acesso às estatísticas.	25
Figura 6 – Página inicial do aluno com informações sobre os instrumentos disponíveis.	25
Figura 7 – Página de aplicação do instrumento de avaliação.	26
Figura 8 – Página de desempenho final do aluno.	26
Figura 9 – CCI's de todos itens.	29
Figura 10 – FII's de todos os itens.	29
Figura 11 – Correlação entre Resolução de Problemas e Programação.	33
Figura 12 – Correlação entre Pensamento Abstrato e Programação.	34
Figura 13 – Correlação entre Raciocínio Matemático e Programação.	36
Figura 14 – Correlação entre Flexibilidade Cognitiva e Programação.	37

LISTA DE TABELAS

Tabela 1 – Consistência Interna do Instrumento de Aptidão em Programação.	28
Tabela 2 – Instrumento de Aptidão em Programação - Todos os Itens Calibrados. . . .	28
Tabela 3 – Teste de Normalidade de Estimativas de Habilidades entre Resolução de Problemas e Programação.	32
Tabela 4 – Teste de Normalidade de Estimativas de Habilidades entre Pensamento Abstrato e Programação.	34
Tabela 5 – Teste de Normalidade de Estimativas de Habilidades entre Raciocínio Matemático e Programação.	35
Tabela 6 – Teste de Normalidade de Estimativas de Habilidades entre Flexibilidade Cognitiva e Programação.	36

LISTA DE ABREVIATURAS E SIGLAS

2ML	Modelo Logístico de 2 parâmetros
CCI	Curva Característica do Item
CS1	Curso Introdutório de Programação
EAP	Expected A Posteriori
FII	Função de Informação do Item
TCT	Teoria Clássica dos Testes
TRI	Teoria de Resposta ao Item

SUMÁRIO

1	INTRODUÇÃO	13
1.1	Contextualização do Problema	13
1.2	Objetivos	14
1.3	Questão de Pesquisa	14
1.4	Estrutura do trabalho	15
2	TEMAS E TRABALHOS RELACIONADOS	16
2.1	Cursos Introdutórios de Programação	16
2.2	Habilidade Predictoras de Programação	16
2.2.1	<i>Resolução de Problemas</i>	<i>16</i>
2.2.2	<i>Pensamento Abstrato</i>	<i>17</i>
2.2.3	<i>Raciocínio Matemático</i>	<i>18</i>
2.2.4	<i>Flexibilidade Cognitiva</i>	<i>19</i>
2.3	Teorias de Mensuração	19
2.3.1	<i>Teoria Clássica de Testes</i>	<i>19</i>
2.3.1.1	<i>Coeficiente de ponto bisserial</i>	<i>20</i>
2.3.1.2	<i>Coeficiente Alfa de Cronbach</i>	<i>20</i>
2.3.2	<i>Teoria de Resposta ao Item</i>	<i>20</i>
3	INSTRUMENTOS PARA MEDIR HABILIDADES PREDITORAS DE PROGRAMAÇÃO E ESCRITA/LEITURA DE CÓDIGO	23
3.1	Screening Programming	23
3.2	Instrumento de Aptidão em Programação	26
3.2.1	<i>Construção do Banco de Itens</i>	<i>26</i>
3.2.2	<i>Calibração do Banco de Itens</i>	<i>27</i>
4	METODOLOGIA	30
4.1	Métrica	30
4.2	Seleção dos Participantes	30
4.3	Preparação	30
4.4	Análise dos Dados	30
4.5	Análise às Ameaças	31
4.6	Execução da Pesquisa	31
5	ANÁLISE E RESULTADOS	32
5.1	Existe correlação entre a habilidade de Resolução de Problemas e Pro- gramação?	32

5.2	Existe correlação entre a habilidade de Pensamento Abstrato e Programação?	33
5.3	Existe correlação entre a habilidade de Raciocínio Matemático e Programação?	35
5.4	Existe correlação entre a habilidade de Flexibilidade Cognitiva e Programação?	36
6	CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS	38
	REFERÊNCIAS	40
	ANEXO A – TERMO DE ASSENTIMENTO	43
	ANEXO B – TERMO DE CONSENTIMENTO LIVRE ESCLARECIDO	44
	APÊNDICE A – INSTRUMENTO DE APTIDÃO EM PROGRAMAÇÃO	45

1 INTRODUÇÃO

Neste Capítulo, apresenta-se uma visão geral desta pesquisa, de modo a descrever a contextualização do problema, objetivos e questões de pesquisas.

1.1 Contextualização do Problema

O Ensino Superior e Técnico enfrenta o desafio de reduzir os índices de falha e evasão dos alunos em seus cursos. Este problema gera prejuízos para o país, atingindo inúmeras instituições e existe uma preocupação clara dos governos e das instituições em atenuar tais índices (INEP, 2013). Essas instituições oferecem a cada ano um crescente número de vagas para novos alunos ingressarem em seus cursos. No entanto, boa parte dos alunos que entram nestes cursos não concluem. A conquista por uma vaga em uma instituição pública seguido do abandono tornou-se um problema generalizado, independente da instituição.

Em Ciência Computação, os índices de evasão e reprovação chegam a ser piores. Logo no primeiro semestre, os alunos sentem dificuldade principalmente na disciplina de Introdução a Programação (CS1). Cerca de 1/3 desses alunos evadem ou falham somente nesta disciplina (WATSON; LI, 2014). Estratégias eficazes de previsão e mitigação de falhas e evasão em CS1 podem fornecer enormes ganhos na resiliência do ensino em programação em larga escala.

Pensando nisso, em parceria com o Laboratório de Neuropsicologia Cognitiva e Inovação Tecnológica da Universidade Federal de Campina Grande, o Laboratório de Práticas de Software e a Universidade Estadual da Paraíba foi desenvolvido o Screening Programming¹ destinado a rastrear habilidades preditoras em programação (ANDRADE, 2022) (DANTAS, 2022). Trata-se de um sistema que apoia os educadores de cursos de Ensino Superior e Técnico da área de informática a mensurar habilidades preditoras de programação (ANDRADE, 2022).

Resultados anteriores mostram que o Screening Programming apresenta itens que contemplam as habilidades preditoras de programação. Tais itens possuem uma boa análise de conteúdo, boas propriedades psicométricas e boa confiabilidade em avaliações profissionais (DANTAS, 2022). Após a instrumentalização, os usuários avaliaram a sua usabilidade e obtiveram uma empatia pelo rastreamento/evolução das habilidades preditoras de programação feito por meio da versão informatizada (ANDRADE, 2022). Esses ganhos vieram da migração por meio das ações inteligentes para mensurar habilidades de forma adaptativa, em vez de avaliar de forma tradicional. Porém, esses estudos não investigaram a relação das habilidades preditoras de programação com habilidades de leitura/escrita de código.

Na literatura, há muitos trabalhos que reforçam que as habilidades de resolução de problemas (LISHINSKI et al., 2016), pensamento abstrato (PARK; HYUN, 2014) (PARK; HYUN; HEUILAN, 2015), raciocínio matemático (ATTALLAH; ILAGURE; CHANG, 2018) (SOUZA et al., 2019) e flexibilidade cognitiva (DURAK, 2020) são precursores de escrita/leitura de

¹ Disponível em: <https://www.screeningprogramming.com/>

código. Com objetivo de contribuir para o estado-da-arte, procura-se nesta pesquisa avançar nas discussões e ter comprovações científicas de que tais habilidades são precursoras de programação.

1.2 Objetivos

A presente pesquisa teve como objetivo principal investigar a relação de um conjunto de habilidades preditoras de programação e a escrita/leitura de códigos em iniciantes em programação. Para alcançar o objetivo geral desta pesquisa, foram necessários atingir os seguintes objetivos específicos:

- Desenvolver itens com boa análise de conteúdo que contemplem os indicadores de habilidades de leitura e escrita de códigos, compondo o Teste de Aptidão em Programação;
- Calibrar o banco de itens em termos das informações que fornecem a respeito do construto psicológico específico avaliado;
- Avaliar a confiabilidade do instrumento por meio das teorias de mensuração;
- Aplicar o Screening Programming e o Instrumento de Aptidão em Programação a um grupo de ingressantes em CS1 a fim de verificar se existe uma relação entre as habilidades preditoras de programação (resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva) e as habilidades de escrita e leitura de códigos.

1.3 Questão de Pesquisa

Esta pesquisa foi dividida em duas etapas, a saber:

- **Construção do Instrumento de Aptidão em Programação.** Desenvolveu-se um banco de itens que avalia habilidades de leitura e escrita de código em programação (Python) e foi avaliada a análise de conteúdo desses itens, bem como as propriedades psicométricas e a confiabilidade do instrumento desenvolvido.
 - **QP1.** Os itens que contemplem os indicadores do Instrumento de Aptidão em Programação possuem boa análise de conteúdo?
 - **QP2.** Os itens do Instrumento de Aptidão em Programação possuem boas propriedades psicométricas?
 - **QP3.** O Instrumento de Aptidão em Programação, a partir do banco de itens, apresenta boa confiabilidade?
- **Correlação entre os Cenários Observados.** Correlacionou-se o desempenho dos sujeitos no Instrumento de Aptidão em Programação e as habilidades preditoras de programação presentes no Screening Programming.

- **QP4.** Existe correlação entre a habilidade Resolução de Problemas e a capacidade de leitura e escrita de código em alunos iniciantes em programação?
- **QP5.** Existe correlação entre a habilidade Pensamento Abstrato e a capacidade de leitura e escrita de código em alunos iniciantes em programação?
- **QP6.** Existe correlação entre a habilidade Raciocínio Matemático e a capacidade de leitura e escrita de código em alunos iniciantes em programação?
- **QP7.** Existe correlação entre a habilidade Flexibilidade Cognitiva e a capacidade de leitura e escrita de código em alunos iniciantes em programação?

1.4 Estrutura do trabalho

Este trabalho apresenta seis capítulos e está organizado da seguinte maneira: no Capítulo 1, apresenta-se uma visão geral desta investigação com relação a contextualização do problema, objetivos e questões de pesquisa; no Capítulo 2, apresenta-se os temas e trabalhos relacionados à pesquisa; no Capítulo 3, apresenta-se o instrumento Screening Programming para medir habilidades preditoras de programação e o desenvolvimento do Teste de Aptidão em Programação para mensurar habilidades de leitura e escrita de código; no Capítulo 4, apresenta-se a metodologia desta pesquisa; no Capítulo 5, analisa-se e discute-se os resultados; no Capítulo 6, apresenta-se as considerações finais e sugestões para trabalhos futuros; e ao final, encontra-se as referências e os apêndices utilizados do decorrer desta pesquisa.

2 TEMAS E TRABALHOS RELACIONADOS

Neste Capítulo, apresenta-se o embasamento teórico a partir de várias áreas e trabalhos que se relacionam e que permitem caracterizar esta pesquisa.

2.1 Cursos Introdutórios de Programação

Codificação é uma instrução escrita por um programador que segue as regras de sintaxe especificadas pela linguagem de programação (ALSHAYE; TASIR; JUMAAT, 2019). Disciplinas introdutórias em cursos de programação são responsáveis por apresentar aos estudantes os aspectos fundamentais para o entendimento e elaboração de um programa. Desta forma é necessário que seja desenvolvido nos alunos habilidades de programação, pois ter estas habilidades é primordialmente necessário para escrever um programa em qualquer linguagem de programação (ALSHAYE; TASIR; JUMAAT, 2019).

Porém, alunos costumam ter dificuldades com estas disciplinas devido a quantidade de habilidades necessárias para seu entendimento e por serem tão complexas (DURAK, 2018). Pois, além de ser preciso entender a sintaxe de uma linguagem de programação, é preciso dos estudantes a compreensão de estruturas de repetição, condição, variáveis, dentre outros conceitos e, para isso, precisam desenvolver diferentes formas de pensamento.

Com isso é notório significativas taxas de alunos que reprovam em disciplinas dos cursos de graduação de programação, o que termina resultando em altas taxas de evasão em programas de ensino superior nesta área (ATTALLAH; ILAGURE; CHANG, 2018). Logo é preciso que hajam formas de analisar os níveis de habilidades preditoras de programação em estudantes, principalmente àqueles que não tiveram contato anterior com este tipo de disciplina, para que possam ser trabalhados e melhorados de forma precisa às suas dificuldades.

Então, para entender os pontos fortes e fracos de um aluno, é preciso que se avalie suas capacidades em diferentes habilidades. Mas é, primeiramente, necessário entender quais são essas habilidades, como medi-las e como trabalhar com base nos dados obtidos dessa mensuração.

2.2 Habilidade Preditoras de Programação

O desenvolvimento de algumas habilidades é capaz de denotar maiores capacidades para aprendizado e prática da programação de computadores. Neste tópico serão elaboradas reflexões e definições sobre estas habilidades e seu impacto no estudo de programação.

2.2.1 Resolução de Problemas

Um problema existe quando há uma disparidade entre o estado atual e o estado desejado, logo uma solução seria a alocação de recursos disponíveis para alterar o estado atual e reduzir a disparidade entre ele e o estado desejado (MICHAELI; ROMEIKE, 2019). Assim, atingir uma solução requer que exista a capacidade de resolução de problemas, que é o esforço consciente no

processamento controlado da informação que visa identificar, descobrir ou inventar uma solução para um problema (ALSHAYE; TASIR; JUMAAT, 2019).

Desta forma, é importante para todo ser humano o domínio da resolução de problemas, porém é especialmente necessário aqueles envolvidos nos estudos de Ciência da Computação. Esta área, particularmente, requer boa capacidade de resolver problemas (CHAUDHRY; RASOOL, 2012). Esta habilidade ajuda a explicar o sucesso na maioria das atividades difíceis de programação (LISHINSKI et al., 2016).

Os empregadores consideram algumas características desejáveis para a participação efetiva no ambiente de trabalho, tais como: habilidades de comunicação, aptidão ao aprendizado, bem como a capacidade de resolver problemas. Sendo que o conhecimento profissional considerado menos importante entre os empregadores (CHAUDHRY; RASOOL, 2012). Levando isso em consideração, é possível demonstrar a necessidade que tal habilidade seja exercitada nos estudantes de todos os níveis, principalmente para estudantes de Ciências Aplicadas e Engenharia, como Ciência da Computação.

Assim sendo, é sugerida a utilização de diferentes métodos de ensino destas técnicas de solução de problemas. Ferramentas dinâmicas que simplificam a visualização da construção de código, evitando as complexidades da sintaxe e da IDE, como a ferramenta Scratch, podem ser usadas para ensinar princípios de ciência da computação que sustentam o pensamento computacional (ROSE, 2016) (TAHERI; HIDEHIKO; TRIPATHY, 2013).

As dificuldades sentidas pelos estudantes em ramos da programação costumam resultar em altas taxas de reprovação e evasão (MATHEW; MALIK; TAWAFK, 2019). Sendo assim, é preciso que cursos de graduação sejam estruturados de forma a compelir os alunos a desenvolver habilidades de resolução de problemas (CHAUDHRY; RASOOL, 2012).

Entretanto, mesmo um estudante com algum conhecimento nas áreas de ciências da computação e desenvolvimento de software, possui dificuldades em como abordar um problema a fins de desenvolver um algoritmo, alcançar uma solução efetiva e implementar essa ideia com um código bem escrito (MULLER; BUTMAN; BUTMAN, 2017). Logo, deve-se guiar os estudantes a definir, abstrair, gerar soluções, avaliar as melhores opções para então implementar a solução de um problema. A elegância de uma solução é frequentemente achada ao simplificar uma ideia complexa.

Essas habilidades podem ser treinadas, porém não são as únicas habilidades determinantes para um bom desempenho em programação. A forma de se pensar é, também, fundamental para o desenvolvimento de programas. Logo o “pensar sobre pensar”, a metacognição, deve ser avaliado e desenvolvido, para que os alunos se tornem capazes de desenvolver pensamento abstrato.

2.2.2 *Pensamento Abstrato*

O pensamento abstrato habita nas capacidades de aprendizado, julgamento e comportamento das pessoas. Ele é fator relevante para a capacidade de compreensão de programas e

suas estruturas. Assim, o pensamento abstrato é a capacidade de separar problemas e resolvê-los individualmente, construindo uma série de etapas para atingir um objetivo específico (PARK; HYUN, 2014) (STATTER; ARMONI, 2020).

De igual modo, pode-se complementar que pensar de forma abstrata é saber alterar o foco sobre o problema, podendo “alterar a resolução” em cima da questão, achando o grau correto de onde se observar o problema, para assim ignorar os detalhes que não importam para o passo que se dá na lógica da solução. Assim, reduzindo o problema em seções menores, melhora a compreensão de como preparar e do que está acontecendo no algoritmo (STATTER; ARMONI, 2020).

Desta forma pode-se apresentar o exemplo de uma estrutura de iteração. A compreensão deste tipo de estrutura é diretamente proporcional às capacidades de pensamento abstrato de um aluno. Isso significa que para o entendimento de uma estrutura iterativa, quando o nível de pensamento abstrato é alto, a compreensão é alta (PARK; HYUN, 2014).

Esta habilidade pode ser evoluída através do treinamento. Utilização de quebra-cabeças, enigmas e programação em blocos são formas que se mostraram úteis no desenvolvimento de pensamento abstrato em crianças (STATTER; ARMONI, 2020).

Considerando o que foi dito, a habilidade de abstração de problemas e observação da figura maior impactará a habilidade de produção e a qualidade de programas produzidos por estudantes da área. Portanto, ter um alto nível de pensamento abstrato é importante para a habilidade de programação, especialmente em problemas de programação difíceis (PARK; HYUN, 2014) (PARK; HYUN; HEUILAN, 2015).

2.2.3 *Raciocínio Matemático*

A primeira geração de computadores foi idealizada por matemáticos. Logo a conexão entre matemática e programação é devido aos programadores terem de atender a lógica, funções e procedimentos matemáticos para atingir seus objetivos (ATTALLAH; ILAGURE; CHANG, 2018).

Com base nisso, a matemática está ligada à computação de diversas formas: estruturas de repetição, funções, métodos e variáveis são alguns dos conceitos que herdaram raízes matemáticas. Sendo assim, a formação matemática dos alunos pode ser utilizada para identificar possíveis dificuldades durante os cursos de programação além de necessários para um bom desempenho em atividades relacionadas à programação, pois boas notas nas disciplinas de ciências, tecnologia, engenharia e matemática influenciam as notas em cursos de programação (SOUZA et al., 2019) (ATTALLAH; ILAGURE; CHANG, 2018).

Assim, em detalhes, aplicações matemáticas são vistas claramente na programação. Conceitos de matemática discreta fornecem familiaridade com estruturas básicas e permitem que alunos modelem diversos componentes de software com conjuntos matemáticos, strings, funções, relações, sistemas numéricos, teorias matemáticas e assim por diante (DRACHOVA et al., 2015). Bem como conceitos de estatística e álgebra linear são aplicados em inteligência

artificial e processamento de imagem.

Consequentemente, bem como o raciocínio abstrato e a capacidade de resolução de problemas, é preciso que o raciocínio matemático seja exercitado e aprimorado, já que a habilidade matemática é responsável por uma boa parte da variação no desempenho da programação e é a principal disciplina utilizada para promover o pensamento computacional (SOUZA et al., 2019).

2.2.4 Flexibilidade Cognitiva

Além da capacidade de resolver problemas, a flexibilidade cognitiva é, também, uma das capacidades empregadas no treinamento de desenvolvimento de programas (DURAK, 2020). Mas o que seria essa flexibilidade cognitiva?

Pode-se argumentar que ao se deparar com um problema, é possível encontrar mais de uma solução possível que satisfaça a situação. Não basta implementar códigos no processo de desenvolvimento de um programa (DURAK, 2018). Assim, quanto mais complexo o problema, mais formas de solucioná-lo podem existir. A denominada flexibilidade cognitiva, então, é a característica individual primordial para orientar o problema a fins de chegar a uma solução e saída, bem como analisar e avaliar o processo da resolução do problema (DURAK, 2020).

Desta forma, pode-se dizer que a flexibilidade cognitiva é o que é conhecido por “pensar fora da caixa”, procurando soluções distintas e diversas para diferentes tipos de problemas. Também é devido observar e aplicar as estratégias mais eficazes ao tópico em questão, identificando os passos necessários para atingir este fim (DURAK, 2020).

2.3 Teorias de Mensuração

A Teoria Clássica do Teste (TCT) e a Teoria da Resposta ao Item (TRI) (NUNNALLY, 1994) (ARAÚJO et al., 2019) são duas referências utilizadas para construção, validação, e avaliação de instrumentos (PASQUALI, 2017). São teorias que avaliam construções cognitivas. Não são teorias contrárias, mas a TRI é um complemento às limitações da TCT.

2.3.1 Teoria Clássica de Testes

A TCT considera a pontuação total de um instrumento como medida primária da avaliação do desempenho de um indivíduo. Além disso, a TCT usa normas para interpretar as pontuações de um instrumento, e essas normas referenciam os intérpretes e classificam as pontuações, por exemplo, para posicionar um indivíduo que é medido por instrumentos ou comparar a pontuação de dois sujeitos. (ARAÚJO et al., 2019).

A TCT está associado à diferença entre a pontuação do indivíduo não testado e o valor real dessa pontuação (PRIMI, 2012). Já válido, a TCT propõe verificar se o construto mede ou deve medir (PASQUALI, 2017). Por meio do escore, é possível utilizar algumas medidas para avaliar a qualidade dos itens e do Instrumento, como o coeficiente de correlação do ponto bisserial e o coeficiente alfa de Cronbach.

2.3.1.1 *Coefficiente de ponto bisserial*

Em testes educacionais, é possível calcular o coeficiente de correlação entre duas variáveis, uma numérica e outra nominal categórica. Neste caso, a variável categórica possui apenas dois valores possíveis (certo/errado); uma dessas variáveis é chamada de dicotômica. Em seguida, para calcular a correlação entre esta variável e outra variável (numérica), procede-se com o cálculo do coeficiente de Pearson da maneira usual, dadas as hipóteses de normalidade da amostra, que é chamado de coeficiente de correlação de ponto bisserial (BAKER, 2001) .

As estimativas do coeficiente de pontos bisseriados, que são os itens em que, se o avaliado acertar este item, tem mais chances de obter melhores resultados em um instrumento. Essa medida funciona pela seguinte característica, quanto maior o coeficiente, mais forte a correlação daquele item com o escore e indica que aquele item é essencial para o resultado total do Instrumento. Essa medida varia de -1 a 1, e quanto mais próximo de 1, mais discriminativo é o item (ARAÚJO et al., 2019).

2.3.1.2 *Coefficiente Alfa de Cronbach*

Após a construção do Instrumento, uma das formas mais eficientes, em termos de tempo e custo, é verificar sua confiabilidade. A consistência interna mede a confiabilidade do Instrumento. A consistência interna consiste em examinar a homogeneidade dos itens que compõem o Instrumento, ou seja, verificar a magnitude das relações entre os itens e o escore total. Pode-se calcular a consistência interna a partir da pontuação geral do Instrumento e da pontuação de cada item.

Calcula-se essa consistência usando o coeficiente alfa de Cronbach variando de 0 a 1; quanto mais próximo de 1 indica que o Instrumento possui consistência interna adequada (ANDRADE; LAROS; GOUVEIA, 2010). Valores mais próximos de 1 indicam que o Instrumento possui consistência interna adequada. Valores entre 0,70 e 0,80 são considerados aceitáveis, mas com ressalvas. Quando os valores estão abaixo de 0,70, significa que os itens que compõem o Instrumento precisam ser reavaliados pelo pesquisador (ARAÚJO et al., 2019).

2.3.2 *Teoria de Resposta ao Item*

A Teoria de Resposta ao Item (TRI) é um conjunto de modelos matemáticos que calcula a probabilidade de resposta do item por meio da proficiência do aluno (variável não observável) e dos parâmetros que expressam as propriedades do item. Quanto maior a proficiência, maior a probabilidade do aluno acertar o item (ARAÚJO et al., 2019). Os itens podem ser binários (certo ou errado), de múltipla escolha ou politomia, em que o aluno precisa escrever a resposta por meio de uma classificação ordenada.

Pode-se usar a TRI para projetar instrumentos de avaliação educacional, calibração de itens (por valores de parâmetros numéricos) e outros processos de desenvolvimento de instrumentos. A TRI ajusta os dados ao modelo. Portanto, muitas vezes é possível comparar as

habilidades de pessoas diferentes ou da mesma pessoa devido aos parâmetros dos itens (BAKER; KIM, 2017).

Na TRI, um conjunto de variáveis de fatores hipotéticos pode prever o comportamento de um indivíduo em relação a um item. Portanto, uma função matemática crescente monótona, cujo gráfico é chamado de Curva Característica do Item (CCI), depende desses fatores (PASQUALI, 2017).

A CCI fornece informações sobre a probabilidade de cada indivíduo acertar o item (BAKER; KIM, 2017). Cada item possui uma CCI que determina sua qualidade. A CCI é o gráfico que representa a relação entre a habilidade estimada e o desempenho nos itens (ARAÚJO et al., 2019). A CCI é influenciada pelas particularidades da TRI, de acordo com o modelo matemático utilizado.

Diferentes modelos matemáticos dependem dos parâmetros envolvidos no instrumento, a saber: dimensionalidade ou tipo de itens. Este artigo considera o modelo logístico unidimensional de dois parâmetros (2ML). A Equação 2.1 descreve a probabilidade do sujeito com habilidade θ de acertar o item j dependendo da discriminação (a) e da dificuldade (b) do item.

$$P(\theta) = \frac{1}{1 + e^{-a_j(\theta - b_j)}} \quad (2.1)$$

Onde,

θ representa o traço/habilidade latente de um indivíduo;

a_j representa o parâmetro a discriminação do item;

b_j representa o parâmetro de dificuldade do item.

A discriminação do item (parâmetro a) descreve indivíduos com diferentes habilidades quanto às chances de acertar o item, detalhando sujeitos com magnitudes próximas à habilidade avaliada. Este parâmetro varia de 0 (não discriminativo) a 4 (extremamente discriminativo). A dificuldade do item (parâmetro b) refere-se à habilidade necessária para um indivíduo com determinada probabilidade de acertar o item, calculada pela chance de acertar o item por acaso. Este parâmetro varia de -4 (itens fáceis) a +4 (itens difíceis).

O 2ML reproduz uma escala chamada traço ou habilidade latente. O nível gerado é padronizado (média = 0 e SD = 1) e, conforme observado para o parâmetro métrico b , em teoria, essa escala pode variar de -4 a +4. Assim, estimou-se os escores usando a TRI usando um método de estimação (FOX; GLAS, 2001).

Neste trabalho, fez-se o uso de Expected A Posteriori (EAP) (KOLEN; TONG, 2010) (LU; THOMAS; ZUMBO, 2005); cada pessoa recebe a pontuação que melhor identifica sua habilidade na escala. O procedimento EAP estima a habilidade de um examinador. O procedimento EAP é uma média de distribuição posterior e o erro padrão após a aplicação do instrumento; depende da função e dos parâmetros de informação do item. A equação 2.2 define a estimativa do traço latente pelo EAP:

$$\theta_{s+1} = \theta_s + \frac{\sum_{i=1}^n a_i [u_i - P_i \theta_s]}{\sum_{i=1}^n a_i^2 P_i(\theta_s) Q_i(\theta_s)} \quad (2.2)$$

Onde,

θ_s representa a capacidade estimada do examinado dentro de s iterações;

a_i representa o parâmetro de inclinação do item i , $i = 1, 2, \dots, N$;

O procedimento EAP estima a capacidade de um examinador. Esta é uma média da distribuição a posteriori e do erro padrão após a aplicação do Instrumento e depende da Função de Informação do Item (FII) e seus parâmetros.

A FII analisa o quanto um item contém informações psicométricas para medição de habilidades. Sendo estatisticamente definido como a quantidade de informação psicométrica que um item contém em todos os pontos ao longo do continuum do traço latente, ele representa (PASQUALI, 2017).

A FII é uma ferramenta poderosa para análise de itens, permitindo saber não apenas quanta informação um item acumula em um determinado valor de θ , mas também em qual valor de θ o item possui a maior quantidade de informações. A FII tem sido o método de análise de itens mais usado pelos construtores de teste atualmente (BAKER, 2001).

3 INSTRUMENTOS PARA MEDIR HABILIDADES PREDITORAS DE PROGRAMAÇÃO E ESCRITA/LEITURA DE CÓDIGO

Neste Capítulo relataram-se o Screening Programming (ANDRADE, 2022) (DANTAS, 2022) para mensurar habilidades preditoras de programação (Seção 3.1) e o desenvolvimento do Instrumento de Aptidão em Programação (Apêndice A) para mensurar habilidades de escrita/leitura de código (Seção 3.2).

3.1 Screening Programming

No Ensino Tradicional, os professores precisam avaliar o desempenho anterior dos alunos que iniciam o CS1, e os educadores muitas vezes ignoram essa etapa devido ao escopo do Ensino Superior. Essa mensuração ajuda os professores a identificar possíveis falhas ou evasões do aluno, focando em estratégias para resgatar esse aluno com antecedência.

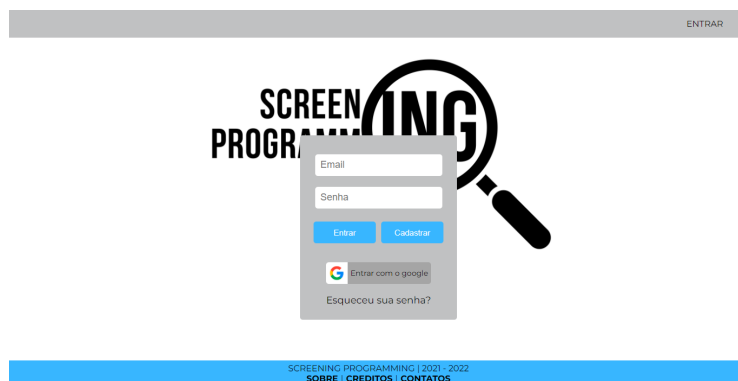
Screening Programming é um sistema que visa auxiliar professores de cursos superiores a coletar e avaliar as habilidades de um indivíduo com base em quatro instrumentos, que mensuram as habilidades preditoras de programação mencionadas neste trabalho. As respostas desses instrumentos são administradas por um algoritmo adaptativo baseado na Teoria de Resposta ao Item (TRI), que consegue avaliar estatisticamente os parâmetros, permitindo que diferentes alunos possam ser comparados entre si. O sistema está disponível em <https://www.screeningprogramming.com/>.

O Screening Programming está estruturado da seguinte forma:

- **Passo 1 - Apresentação:** O sistema inicia com a página inicial e suas funcionalidades, como sobre, créditos, contatos, entrada e cadastro de conta. O usuário pode fazer login se já tiver uma conta cadastrada. Caso contrário, ele precisará se registrar. Ao se cadastrar e efetuar login, o usuário continuará seu cadastro;
- **Etapa 2 - Estrutura das Atividades:** O professor terá acesso para cadastrar/editar/excluir grupos, convidar/remover alunos dos grupos e visualizar os instrumentos de avaliação;
- **Etapa 3 - Mensuração das Habilidades:** o aluno terá acesso para responder aos itens presentes em cada instrumento de avaliação e visualizar seu desempenho;
- **Etapa 4 - Relatório final:** O professor visualizará o relatório com os resultados do desempenho dos alunos no processo de mensuração do instrumento.

A Figura 1 representa a página inicial da Screening Programming. Por meio dela, o usuário pode acessar as páginas sobre créditos (Fig. 2), contatos, cadastro e login.

Figura 1 – Página inicial do Screening Programming.



Fonte: (ANDRADE, 2022).

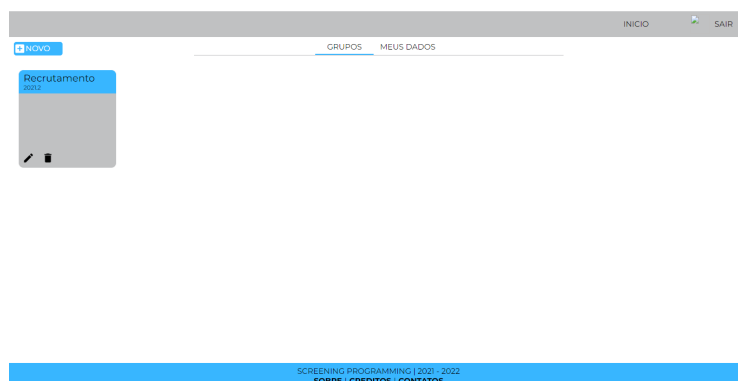
Figura 2 – Página de créditos do Screening Programming.



Fonte: (ANDRADE, 2022).

A Figura 3 ilustra a página inicial do sistema para o Professor com informações sobre os grupos criados. A Figura 4 ilustra a página de acesso ao grupo, onde o professor pode visualizar o relatório do aluno e os instrumentos utilizados para avaliar as habilidades predictoras de programação. Além disso, nesta página, o usuário pode acessar a funcionalidade de editar ou excluir grupos ou excluir alunos do grupo.

Figura 3 – Página inicial do professor com informações sobre os grupos criados.



Fonte: (ANDRADE, 2022).

Figura 4 – Página inicial do grupo.

ID	Matricula	Aluno	Ações
25	21208105	Arthur Pereira	
20	21208156	Wlano	
24	191680527	LUANDERLANDY FELLIPE	
29	212080202	Lukas Christopher de souza	

Fonte: (ANDRADE, 2022).

A Figura 5 ilustra a página de acesso às estatísticas, onde o professor poderá visualizar o relatório geral do grupo por habilidade. Além disso, o usuário pode ter a estatística por item.

Figura 5 – Página de acesso às estatísticas.

Fonte: (ANDRADE, 2022).

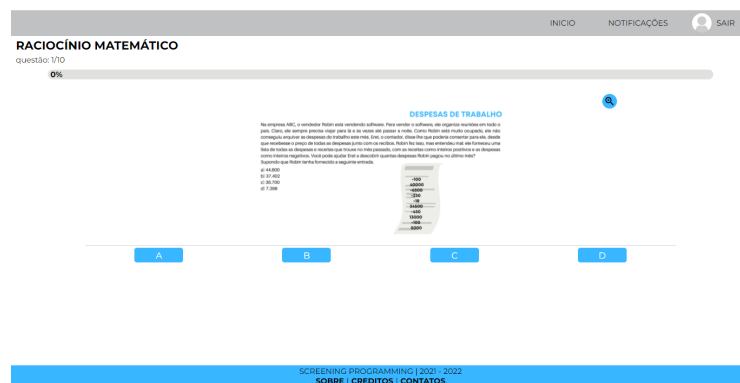
A Figura 6 ilustra a página inicial do sistema para o aluno com a disposição de avaliação dos instrumentos divulgados pelo professor.

Figura 6 – Página inicial do aluno com informações sobre os instrumentos disponíveis.

Fonte: (ANDRADE, 2022).

A Figura 7 ilustra a página inicial da aplicação de um instrumento de avaliação pelo aluno. O aluno responderá cada item até o preenchimento do instrumento. A Figura 8 ilustra a página de desempenho final do aluno, o questionário, suas respostas corretas e sua pontuação em cada habilidade avaliada.

Figura 7 – Página de aplicação do instrumento de avaliação.



Fonte: (ANDRADE, 2022).

Figura 8 – Página de desempenho final do aluno.

ATIVIDADE	ACERTOS(%)	SCORE
RESOLUÇÃO DE PROBLEMAS	100 %	10/10
PENSAMENTO ABSTRATO	70 %	7/10
RACIOCÍNIO MATEMÁTICO	80 %	8/10
FLEXIBILIDADE COGNITIVA	70 %	7/10
GERAL	80 %	32/40

Fonte:(ANDRADE, 2022).

3.2 Instrumento de Aptidão em Programação

Nesta seção discuti-se a construção e calibração do banco de itens, bem como a confiabilidade do Instrumento de Aptidão em Programação (Apêndice A).

3.2.1 Construção do Banco de Itens

Essa etapa consiste em verificar se os itens apresentam boa análise de conteúdo. Responderá-se a seguinte pergunta de pesquisa:

- **QP1.** Os itens que contemplem os indicadores do Instrumento de Aptidão em Programação possuem boa análise de conteúdo?

Junto com a equipe multidisciplinar de especialistas, projetou-se itens fáceis, médios e difíceis de serem resolvidos. Redigiu-se os itens e analisou-se o conteúdo dos itens de forma a garantir que os itens se referem à habilidade que se busca estimar.

Nesta etapa, três professores universitários da comunidade local, especialistas em ensino de programação, participaram do estudo. Os professores avaliaram se os itens presentes no instrumento mensuram o constructo do examinado. Para avaliar o grau de concordância entre os juízes, aplicou-se a função *agree* disponível no pacote *irr* na linguagem *R*. Esta concordância conseguiu atingir 92%. Mas será que é confiável? Para tanto, a fim de avaliar o grau de confiabilidade entre os juízes, aplicou-se o teste estatístico Kappa de Fleiss, com 80% de significância (concordância substancial). Calculou-se por meio da função *kappam.fleiss* presente no pacote *irr* da linguagem *R*. O nível de confiabilidade entre os juízes é de 0,935 (quase perfeita).

Desta forma, pôde-se analisar que os níveis de confiabilidade e concordância entre os juízes é alta. Após a análise das respostas entre os juízes, todos os itens obtiveram índices acima de concordância acima de 80% e foram selecionados para a construção do Instrumento de Aptidão em Programação.

3.2.2 *Calibração do Banco de Itens*

Nesta subseção, apresenta-se e discute-se os resultados da etapa de calibração do banco de itens. Essa etapa consiste em verificar as propriedades psicométricas dos itens do instrumento e analisar se a escala construída está minimamente ajustada para dar continuidade ao estudo. Responderá-se as seguintes perguntas de pesquisa:

- **QP2.** Os itens do Instrumento de Aptidão em Programação possuem boas propriedades psicométricas?
- **QP3.** O Instrumento de Aptidão em Programação, a partir do banco de itens, apresenta boa confiabilidade?

Apresentou-se os itens aos participantes com objetivo de obter informações como acerto/erro do item. Para este estudo, participaram 80 estudantes oriundos do curso de Ciência da Computação da Universidade Estadual da Paraíba, Patos, Paraíba, Brasil. Aplicou-se o instrumento de forma presencial, os alunos tinham um prazo de 2 horas para responder os itens, salientando que a resposta deveria ser feita sem consulta ou com algum monitor/colega da disciplina. Considerou-se as respostas dos alunos durante a aplicação, e as transformou em itens do tipo certo/errado (itens dicotômicos), portanto, na correção atribuindo 0 ao errar e 1 ao acertar.

Analisou-se os dados coletados nesta fase pela TRI com auxílio da ferramenta Excel disponível em: <http://psychometricon.net/libirt/> para análise e ajuste do 2ML pela estimativa marginal de máxima verossimilhança, com finalidade de verificar: i) a consistência interna do instrumento (apresentada na Tabela 1, a partir dos dados obtidos pelo alfa de Cronbach, conclui-

se que o instrumento é confiável) e; ii) a estimativa dos parâmetros dos itens do instrumento que apresenta-se a seguir.

Tabela 1 – Consistência Interna do Instrumento de Aptidão em Programação.

Instrumento	Sujeitos	Itens	Pontuação Média	Desvio Padrão	Cronbach Alpha
Teste de Aptidão em Programação	80	8	4,269	2,146	0,767

Fonte: Elaborada pelo autor (2022).

Interpretou-se a distribuição das respostas dos alunos em cada item por meio do 2ML, além disso foi considerada a proporção de acertos e a correlação ponto bisserial entre a resposta correta no item e a pontuação total no instrumento. Na Tabela 2, apresentou-se os itens que compõem o Instrumento de Aptidão em Programação com os respectivos parâmetros.

Tabela 2 – Instrumento de Aptidão em Programação - Todos os Itens Calibrados.

Id	a	b	Proporção de Acertos	Correlação Ponto Bisserial
Q1	1,814	-0,272	0,577	0,517
Q2	2,353	1,241	0,167	0,472
Q3	1,289	-2,153	0,897	0,311
Q4	3,189	0,384	0,372	0,613
Q5	2,330	-0,291	0,590	0,566
Q6	3,921	-1,423	0,873	0,422
Q7	1,687	0,392	0,410	0,514
Q8	0,844	0,800	0,359	0,338

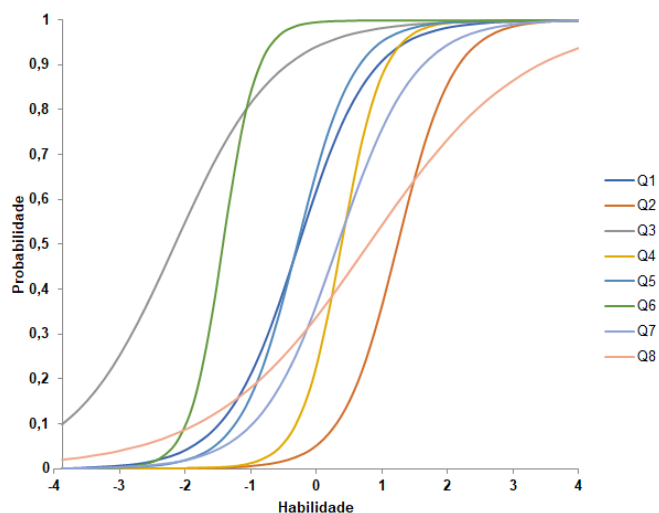
Fonte: Elaborada pelo autor (2022).

Após a constatação do modelo logístico 2ML, não foram encontrados valores críticos para os parâmetros estimados, todos os itens do Instrumento de Aptidão em Programação possuem valores superiores a 0,30 para o índice de discriminação; e para o índice de dificuldade valores entre 3,95 e -3,95. Além disso, os resultados revelaram que este instrumento apresenta itens fáceis (com índices acima de 75%), itens moderados (com índices entre 50 a 75%) e itens difíceis (com índices abaixo de 50%).

As correlações ponto-bisserial revelaram tendência de escolha da opção errada pelos participantes que obtiveram escores mais altos no teste para os itens Q3, Q6 e Q8. Apesar disso, todos os itens se ajustaram adequadamente ao 2ML, logo apresentam boa confiabilidade e bom índice de separação do constructo.

Apresenta-se na Figura 9 a representação gráfica das CCI's dos itens do Instrumento, em que se destacam os valores extremos dos parâmetros discriminação e dificuldade. Pode-se observar pelas CCI's que o item Q6 é o mais discriminativo e Q8 é o menos discriminativo. Já o item Q3 é o mais fácil e o Q2 é o item mais difícil.

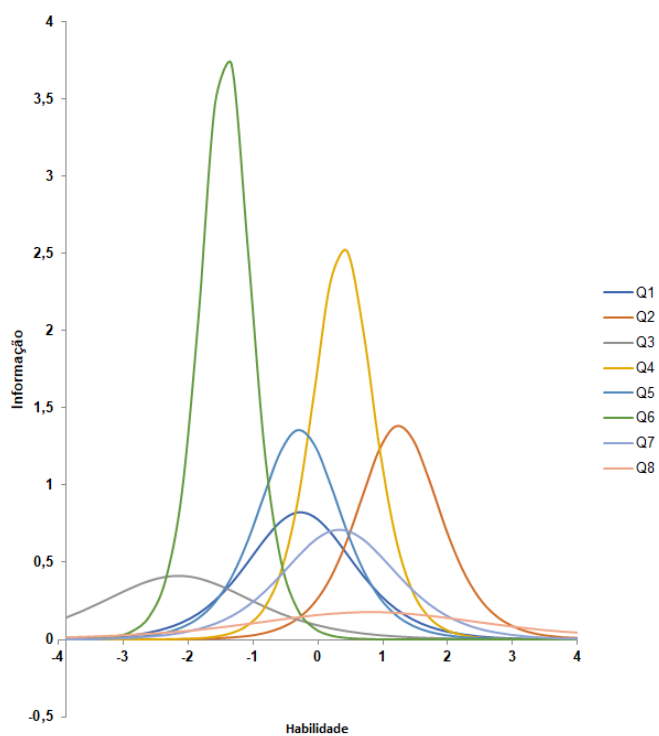
Figura 9 – CCI's de todos itens.



Fonte: Elaborada pelo autor (2022).

Na Figura 10, há a representação gráfica das FII's dos itens do instrumento, em que se destaca o quanto de informação cada item fornece em uma determinada região do traço latente. Pode-se observar que o item Q5 oferece mais informação para avaliar sujeitos de habilidade mediana, ou seja, e um instrumento informatizado, esse item seria o primeiro a ser escolhido, se o instrumento situar a habilidade Theta em 0 como a habilidade média.

Figura 10 – FII's de todos os itens.



Fonte: Elaborada pelo autor (2022).

4 METODOLOGIA

Este Capítulo apresentará o planejamento do estudo que visa verificar a relação entre um conjunto de habilidades preditoras de programação e a escrita/leitura de códigos em iniciantes em programação.

4.1 Métrica

A métrica correlação é amplamente utilizada em pesquisas para avaliar a dependência de duas variáveis. Neste estudo, procura-se verificar se existe uma relação entre as habilidades preditoras de programação (resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva) com a escrita/leitura de códigos em iniciantes em programação.

4.2 Seleção dos Participantes

Este estudo foi composto por 80 (oitenta) participantes, que atenderam aos seguintes critérios de inclusão:

- O participante assinou o Termo de Assentimento (Anexo A) ou o Termo de Consentimento Livre Esclarecido (Anexo B) e foi informado, de forma simplificada, sobre os procedimentos realizados;
- O participante estava matriculado no primeiro período de um curso de graduação ou curso técnico na área de Ciência da Computação;
- O participante não podia apresentar limitações sensoriais, cognitivas ou queixas auditivas e visuais.

4.3 Preparação

Para garantir a qualidade e o acesso dos participantes nos processos experimentais, modularizou-se as ações de forma virtual. Para isso, foi necessário que o participante tivesse acesso um dispositivo móvel ou computador com acesso à Internet para fornecer respostas aos instrumentos de avaliação.

4.4 Análise dos Dados

Analisou-se os dados coletados por meio dos testes psicométricos de acordo com a estatística descritiva e inferencial, visando atender aos objetivos propostos do estudo.

4.5 Análise às Ameaças

Foram considerados alguns fatores que podem gerar ameaças e influenciar diretamente nas conclusões deste trabalho. Entre eles:

- Problemas relacionados à interpretação incorreta das questões;
- Os participantes da pesquisa podem se sentir intimidados ou desconfortáveis ao fazer os testes. Aplicou-se as diretrizes do comitê de ética em pesquisa para minimizar essa possível restrição. O Comitê de Ética em Pesquisa com Seres Humanos da Universidade Federal de Campina Grande e da Universidade Estadual da Paraíba aprovaram esta pesquisa (Protocolos: 23933919.4.0000.5182 | 23933919.4.3001.5187). Somente os participantes que assinaram o Termo de Assentimento (Anexo A) ou o Termo de Consentimento Livre Esclarecido (Anexo B) participaram deste estudo;
- Os instrumentos do Screening Programming foram corrigidos virtualmente para mitigar possíveis erros humanos;
- Como toda pesquisa empírica, este trabalho apresenta ameaças à validade. O número de sujeitos participantes do estudo não permite a generalização dos resultados;
- A amostra deve ser considerada, pois permite a formação de um banco de dados baseado em probabilidade, estatística, axiomas de medição e considerando o objetivo do instrumento. Este banco deve ter o controle centralizado do aplicativo.

4.6 Execução da Pesquisa

Para verificar a correlação entre as variáveis, foram adotadas as seguintes etapas:

- Aplicar o Instrumento de Aptidão em Programação para mensurar a escrita/leitura de códigos de 80 (oitenta) iniciantes em programação. Os participantes tiveram 2 horas para responder as questões presentes no instrumento.
- Separar os sujeitos anteriores em 4 grupos e utilizar o Screening Programming para mensurar as habilidades preditoras de programação. Mensurar as habilidades de resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva, respectivamente. Os participantes responderam ao banco de questões entre 2 e 4 horas.
- Estimar as habilidades dos sujeitos no Teste de Aptidão em Programação por meio marco eirt adotando o 2ML pela estimativa marginal de máxima verossimilhança;
- Correlacionar a estimativa das habilidades dos sujeitos no Teste de Aptidão em Programação com as habilidades preditoras estimadas pelo algoritmo adaptativo geradas pelo Sistema Screening Programming.

5 ANÁLISE E RESULTADOS

Neste Capítulo, apresenta e discuti-se os resultados desta pesquisa, que visa investigar se um conjunto de habilidades preditoras de programação se relacionam com a escrita e a leitura de códigos em iniciantes.

Na plataforma Screening Programming avalia-se as 4 (quatro) habilidades preditoras de programação: i) Resolução de Problemas, ii) Pensamento Abstrato, iii) Raciocínio Matemático e iv) Flexibilidade Cognitiva. Correlacionando o desempenho dos sujeitos em cada uma dessas habilidades com o seu desempenho no Teste de Aptidão em Programação.

5.1 Existe correlação entre a habilidade de Resolução de Problemas e Programação?

Primeiramente calcula-se a distribuição normal dos dados através da função *shapiro.test()* presente na linguagem R. Assim apresenta-se, na Tabela 3, os resultados do teste de normalidade das estimativas de habilidades entre os cenários observados.

Tabela 3 – Teste de Normalidade de Estimativas de Habilidades entre Resolução de Problemas e Programação.

Hipótese Nula	p-value
A estimativa da habilidade em Programação pelo Instrumento de Aptidão em Programação não segue uma distribuição normal.	0,09932
A estimativa da habilidade em Resolução de Problemas pelo Screening Programming não segue uma distribuição normal.	0,06143

Fonte: Elaborada pelo autor (2022).

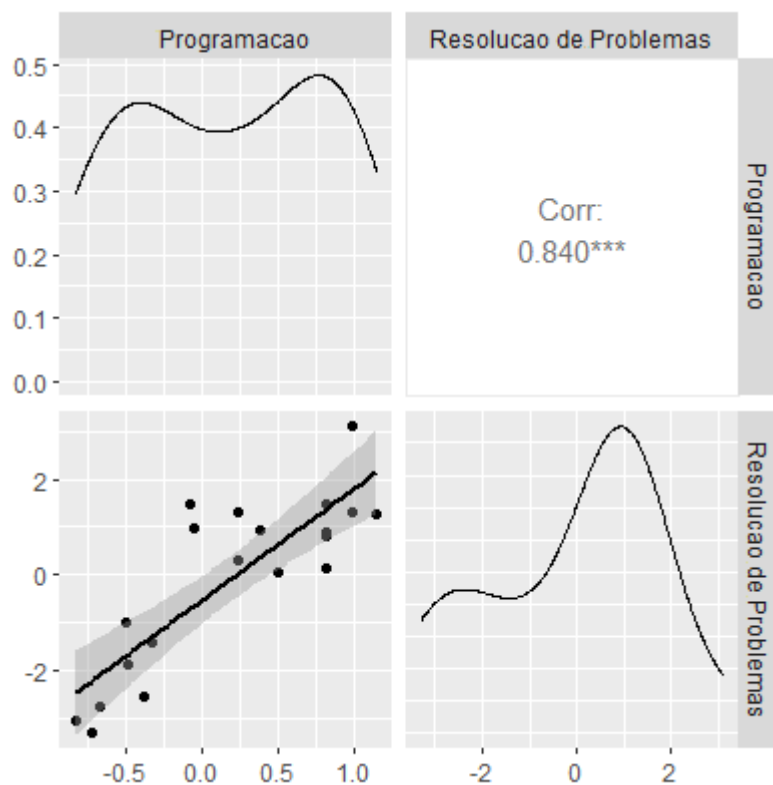
Como os dados seguem uma distribuição normal, obteve-se a correlação de Pearson entre as variáveis e a plotagem dos gráficos presentes na Figura 11 por meio da biblioteca *GGally* na linguagem R, utilizando a função *ggpairs()*. Conforme se pode analisar, existe uma correlação forte entre as variáveis resolução de problemas e programação (leitura e escrita de códigos).

Estes resultados são reforçados em outros estudos (LISHINSKI et al., 2016) (KOULOURI; LAURIA; MACREDIE, 2014) que correlacionaram a capacidade de resolver problemas significativamente com o desempenho em atribuições de programação. Como resolução de problemas requer o esforço consciente do sujeito no processamento controlado da informação, o sujeito consegue identificar/descobrir/inventar uma solução para um problema (ALSHAYE; TASSIR; JUMAAT, 2019), utilizando para tanto, alguma sintaxe de uma linguagem de programação. De tal forma que, a habilidade de resolução de problemas produz um benefício real e quantificável para a habilidade de programação dos alunos (KOULOURI; LAURIA; MACREDIE, 2014).

Desta forma, a habilidade de resolução de problemas é considerada parte integrante da compreensão dos conceitos precisos do domínio de programação para iniciantes em CS1 (MALIK, 2016). Não somente nesta área, mas vários campos usam a resolução de problemas

como uma habilidade essencial para o desenvolvimento profissional; portanto, é útil ter alguma familiaridade básica com a resolução de problemas, independentemente do trabalho ou estudo (TAHERI; HIDEHIKO; TRIPATHY, 2013).

Figura 11 – Correlação entre Resolução de Problemas e Programação.



Fonte: Elaborada pelo autor (2022).

A falta de habilidade para resolver problemas é considerada uma das principais deficiências enfrentadas pelos novatos, é exacerbada pela sintaxe da linguagem que os novatos usam (HOOSHYAR et al., 2015). Os participantes, no geral, que obtiveram estimativas baixas no Screening Programming em resolução de problemas também obtiveram desempenho fraco no Teste de Aptidão em Programação. Estes participantes poderiam ser monitorados pelo professor para mitigar prováveis evasões ou fracassos na disciplina, pois a habilidade resolução de problema pode ser estimulada com exercícios e treinamentos (CHAUDHRY; RASOOL, 2012).

5.2 Existe correlação entre a habilidade de Pensamento Abstrato e Programação?

Calcula-se a distribuição normal desses dados através da função *shapiro.test()* presente na linguagem R. Na Tabela 4 são apresentados os resultados do teste de normalidade das estimativas de habilidades entre os cenários observados.

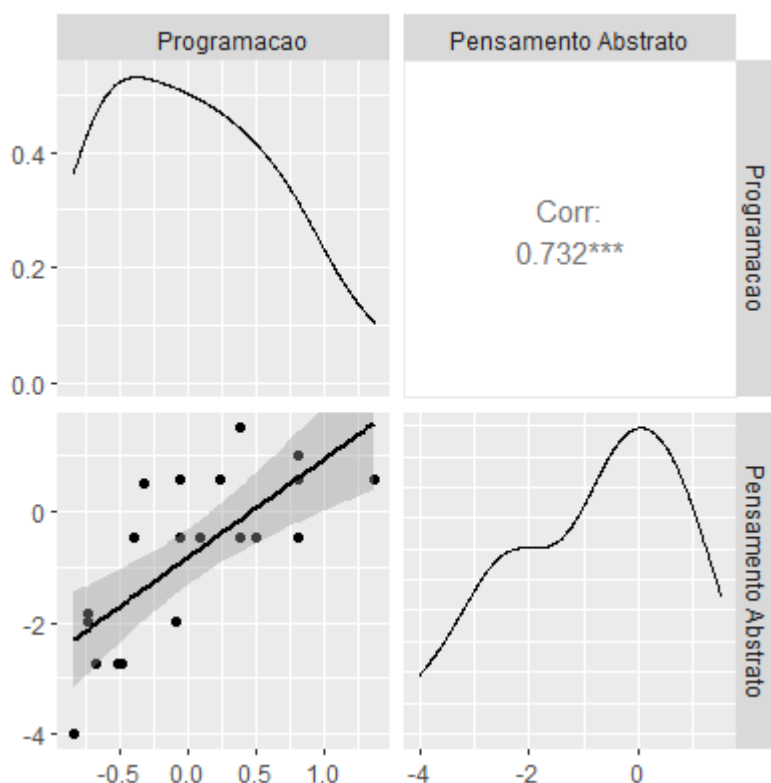
Tabela 4 – Teste de Normalidade de Estimativas de Habilidades entre Pensamento Abstrato e Programação.

Hipótese Nula	p-value
A estimativa da habilidade em Programação pelo Instrumento de Aptidão em Programação não segue uma distribuição normal.	0,1114
A estimativa da habilidade em Pensamento Abstrato pelo Screening Programming não segue uma distribuição normal.	0,3556

Fonte: Elaborada pelo autor (2022).

Como os dados seguem uma distribuição normal, obteve-se a correlação de Pearson entre as variáveis e a plotagem dos gráficos presentes na Figura 12 por meio da biblioteca *GGally* na linguagem *R*, utilizando a função *ggpairs()*. Conforme se pode analisar, existe uma correlação forte entre as variáveis pensamento abstrato e programação (leitura e escrita de códigos).

Figura 12 – Correlação entre Pensamento Abstrato e Programação.



Fonte: Elaborada pelo autor (2022).

Efeito moderador na habilidade de programar entre o nível de familiaridade com programação e pensamento abstrato também foi relatado em estudos recentes (PARK; HYUN, 2014) (PARK; HYUN; HEUILAN, 2015). Percebe-se que quando os alunos têm um alto nível de habilidades de pensamento abstrato, suas habilidades de programação melhoram muito mais do que aqueles com um baixo nível de habilidades de pensamento abstrato. É Observável este padrão de comportamento com a aplicação do Screening Programming e o Teste de Aptidão em

Programação. Nas questões de programação em que envolviam um nível alto de pensamento abstrato dos alunos (por exemplo, os itens Q2 e Q3), alunos que obtiveram desempenho fraco em pensamento abstrato também errou esses itens.

O pensamento abstrato influencia a maneira como as pessoas aprendem informações, formam julgamentos e regulam o comportamento. Por exemplo, o item Q8 necessita do conhecimento em estruturas de iteração, o pensamento abstrato pode afetar diretamente a forma do aluno em compreender um programa que usa a estrutura de iteração.

Se o nível de pensamento abstrato do aluno for mais alto, ele compreenderá melhor o programa. Quando o professor fornece ao aluno um problema de programação descomplicado, a habilidade de pensamento abstrato não é discriminatória na avaliação dos alunos. Porém, quando um problema de programação é difícil, o alto nível de capacidade de pensamento abstrato desempenha um papel essencial na solução do problema (PARK; HYUN, 2014).

5.3 Existe correlação entre a habilidade de Raciocínio Matemático e Programação?

Calcula-se a distribuição normal desses dados através da função *shapiro.test()* presente na linguagem *R*. Na Tabela 5 são apresentados os resultados do teste de normalidade das estimativas de habilidades entre os cenários observados.

Tabela 5 – Teste de Normalidade de Estimativas de Habilidades entre Raciocínio Matemático e Programação.

Hipótese Nula	p-value
A estimativa da habilidade em Programação pelo Instrumento de Aptidão em Programação não segue uma distribuição normal.	0,05805
A estimativa da habilidade em Raciocínio Matemático pelo Screening Programming não segue uma distribuição normal.	0,4409

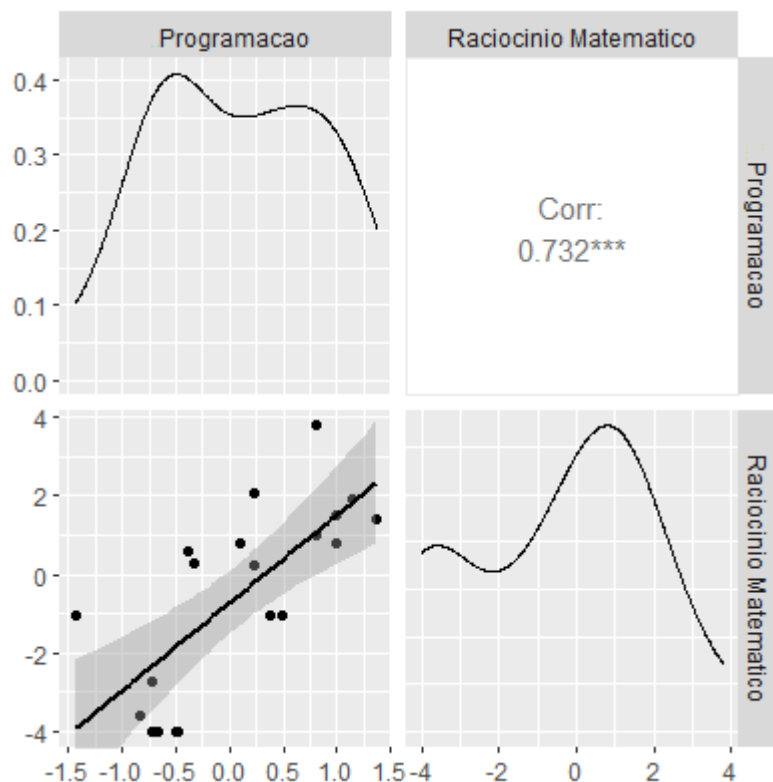
Fonte: Elaborada pelo autor (2022).

Como os dados seguem uma distribuição normal, obtive-se a correlação de Pearson entre as variáveis e a plotagem dos gráficos presentes na Figura 13 por meio da biblioteca *GGally* na linguagem *R*, utilizando a função *ggpairs()*. Conforme se pode analisar, existe uma correlação forte entre as variáveis raciocínio matemático e programação (leitura e escrita de códigos).

É normal que as habilidades de programação sejam frequentemente associadas às habilidades de resolução de problemas, e certos tipos de habilidades matemáticas afetam as habilidades analíticas que contribuem para a experiência de aprender a programar no computador. Há uma correlação entre habilidades matemáticas e potencial de programação, como aponta outros estudos (SOUZA et al., 2019). Os alunos consideram a lógica da programação e da metodologia de ensino como as principais fontes de dificuldades. Além da sintaxe e semântica de uma linguagem de programação, pensamento lógico e matemático, habilidades de numeramento e visualização, conhecimento de álgebra e cálculo, conhecimento de funções condicionais e recursivas são essen-

ciais para projetar funções e procedimentos durante a atividade de programação (ATTALLAH; ILAGURE; CHANG, 2018).

Figura 13 – Correlação entre Raciocínio Matemático e Programação.



Fonte: Elaborada pelo autor (2022).

5.4 Existe correlação entre a habilidade de Flexibilidade Cognitiva e Programação?

Calcula-se a distribuição normal desses dados através da função *shapiro.test()* presente na linguagem *R*. Na Tabela 6 são apresentados os resultados do teste de normalidade das estimativas de habilidades entre os cenários observados.

Tabela 6 – Teste de Normalidade de Estimativas de Habilidades entre Flexibilidade Cognitiva e Programação.

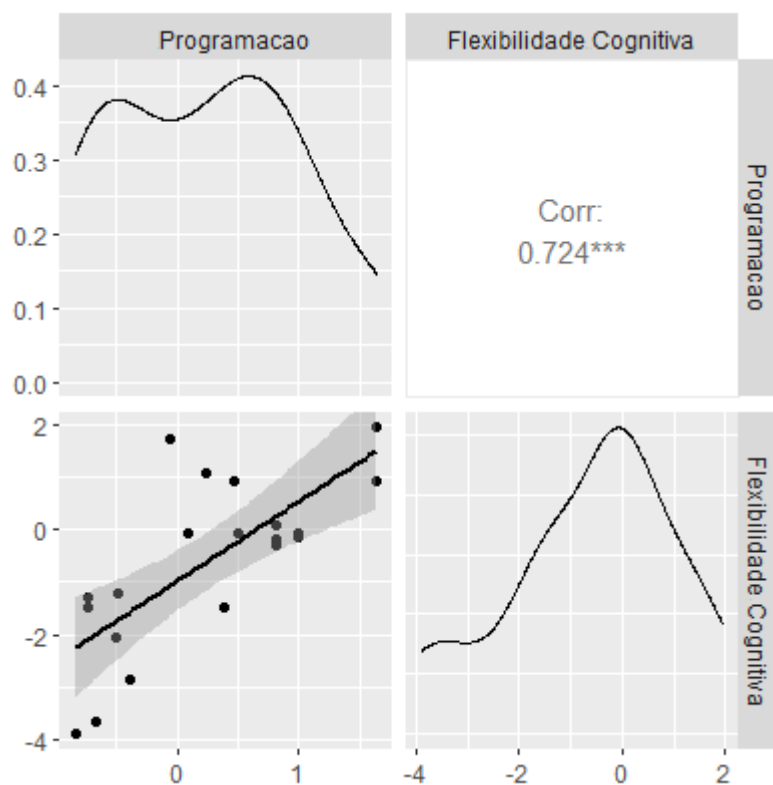
Hipótese Nula	p-value
A estimativa da habilidade em Programação pelo Instrumento de Aptidão em Programação não segue uma distribuição normal.	0,338
A estimativa da habilidade em Flexibilidade Cognitiva pelo Screening Programming não segue uma distribuição normal.	0,1863

Fonte: Elaborada pelo autor (2022).

Como os dados seguem uma distribuição normal, obtive-se a correlação de Pearson entre as variáveis e a plotagem dos gráficos presentes na Figura 14 por meio da biblioteca *GGally* na

linguagem *R*, utilizando a função *ggpairs()*. Conforme se pode analisar, existe uma correlação forte entre as variáveis flexibilidade cognitiva e programação (leitura e escrita de códigos), conforme também aponta em estudos recentes (DURAK, 2018) (DURAK, 2020).

Figura 14 – Correlação entre Flexibilidade Cognitiva e Programação.



Fonte: Elaborada pelo autor (2022).

A flexibilidade cognitiva permite ao aluno empregar as estratégias de aprendizagem mais eficazes relacionadas ao tema em estudo ou identificar as etapas para resolver um problema, encontrar outras soluções, controlar o processo de aprendizagem e controlar os produtos ou oportunidades de autorregulação. Desta forma, os professores devem desenvolver esta habilidade a fim de melhorar o desempenho de programação dos alunos (DURAK, 2020).

6 CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS

Foi construído um Instrumento de Aptidão em Programação a fim de obter um banco de itens que avalia habilidades de leitura e escrita de código em programação (Python) e foi avaliada a análise de conteúdo desses itens, bem como as propriedades psicométricas e a confiabilidade do instrumento desenvolvido. Os resultados foram:

- **(QP1.)** Os itens presentes no Instrumento de Aptidão em Programação apresentam boa análise de conteúdo que contemplem as habilidades de leitura/escrita de códigos;
- **(QP2.)** Os itens que contemplem o Instrumento de Aptidão em Programação possuem boas propriedades psicométricas;
- **(QP3.)** O Instrumento de Aptidão em Programação, a partir do banco de itens, apresenta boa confiabilidade em avaliações profissionais.

Para entender melhor se existe uma relação entre habilidades preditoras de programação e a escrita/leitura de códigos em iniciantes foi necessário aplicar instrumentos com validade psicométrica. Para tanto, avaliou-se as habilidades preditoras de programação por meio do Screening Programming e foi estimada a habilidade de escrita/leitura de código por meio do Teste de Aptidão em Programação. Correlacionou-se o desempenho dos sujeitos e foram obtidos os seguintes resultados em cada questão específica dentro o escopo mais amplo:

- **(QP4.)** Existe uma correlação forte entre a habilidade Resolução de Problemas e a capacidade de leitura e escrita de código em alunos iniciantes em programação.
- **(QP5.)** Existe uma correlação forte entre a habilidade Pensamento Abstrato e a capacidade de leitura e escrita de código em alunos iniciantes em programação.
- **(QP6.)** Existe uma correlação forte entre a habilidade Raciocínio Matemático e a capacidade de leitura e escrita de código em alunos iniciantes em programação.
- **(QP7.)** Existe uma correlação forte entre a habilidade Flexibilidade Cognitiva e a capacidade de leitura e escrita de código em alunos iniciantes em programação.

Mediante dos resultados, pretende-se realizar novos estudos. Entre as diversas possibilidades, destacam-se:

- Informar o Teste de Aptidão em Programação em sua versão adaptativa;
- Realizar um experimento a fim de verificar se, a partir da escala construída e do banco de itens, o sistema adaptativo prevê o levantamento de características que possam identificar sujeitos com fraca/forte habilidade em tarefas preditoras de programação;

- Verificar o impacto do instrumento em relação a diminuição das taxas de evasão e reprovação em CS1.

REFERÊNCIAS

- ALSHAYE, I.; TASIR, Z.; JUMAAT, N. F. The conceptual framework of online problem-based learning towards problem-solving ability and programming skills. In: IEEE, PULAU PINANG, MALAYSIA. *Proceedings of the Conference on e-Learning, e-Management e-Services (IC3e)*. [S.l.], 2019. Citado 3 vezes nas páginas 16, 17 e 32.
- ANDRADE, J. M.; LAROS, J. A.; GOUVEIA, V. V. O uso da teoria de resposta ao item em avaliações educacionais: Diretrizes para pesquisadores. *Avaliação Psicológica*, Instituto Brasileiro de Avaliação Psicológica, v. 9, n. 3, 2010. Citado na página 20.
- ANDRADE, P. H. Análise da usabilidade de um sistema web na mensuração de habilidades cognitivas predictoras em novatos em programação. In: *Trabalho de Conclusão de Curso, Monografia (Bacharelado em Computação) – Centro de Ciências e Tecnologias, Universidade Estadual da Paraíba*. [S.l.: s.n.], 2022. Citado 5 vezes nas páginas 13, 23, 24, 25 e 26.
- ARAÚJO, A. L. S. O. et al. Metodologia de pesquisa em informática na educação: Abordagem quantitativa de pesquisa. In: _____. Porto Alegre: SBC, 2019. cap. Teoria de Resposta ao Item. Citado 3 vezes nas páginas 19, 20 e 21.
- ATTALLAH, B.; ILAGURE, Z.; CHANG, Y. K. The impact of competencies in mathematics and beyond on learning computer programming in higher education. In: IEEE, DUBAI, UNITED ARAB EMIRATES. *Proceedings of the Information Technology Trends (ITT)*. [S.l.], 2018. Citado 4 vezes nas páginas 13, 16, 18 e 36.
- BAKER, F. B. *The Basics of Item Response Theory*. [S.l.]: ERIC, 2001. Citado 2 vezes nas páginas 20 e 22.
- BAKER, F. B.; KIM, S.-H. *The basics of item response theory using R*. [S.l.]: Springer, 2017. Citado na página 21.
- CHAUDHRY, N.; RASOOL, G. A case study on improving problem solving skills of undergraduate computer science students. *World Applied Sciences Journal*, Springer, Pakistan, v. 20, n. 1, 2012. Citado 2 vezes nas páginas 17 e 33.
- DANTAS, D. M. Análise da confiabilidade de um instrumento na mensuração de habilidades cognitivas predictoras em novatos em programação. In: *Trabalho de Conclusão de Curso, Monografia (Bacharelado em Computação) – Centro de Ciências e Tecnologias, Universidade Estadual da Paraíba*. [S.l.: s.n.], 2022. Citado 2 vezes nas páginas 13 e 23.
- DRACHOVA, S. V. et al. Teaching mathematical reasoning principles for software correctness and its assessment. *ACM Transactions on Computing Education (TOCE)*, ACM New York, NY, United States, v. 15, n. 3, 2015. Citado na página 18.
- DURAK, H. Y. The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technology, Knowledge and Learning*, Springer, 2018. Citado 3 vezes nas páginas 16, 19 e 37.
- DURAK, H. Y. Modeling different variables in learning basic concepts of programming in flipped classrooms. *Journal of Educational Computing Research*, v. 58, n. 1, 2020. Citado 3 vezes nas páginas 13, 19 e 37.

FOX, J. P.; GLAS, C. A. W. Bayesian estimation of a multilevel irt model using gibbs sampling. *Psychometrika*, Springer, v. 66, n. 2, 2001. Citado na página 21.

HOOSHYAR, D. et al. Flowchart-based bayesian intelligent tutoring system for computer programming. In: IEEE, KUALA LUMPUR, MALAYSIA. *Proceedings of the International Conference on Smart Sensors and Application (ICSSA)*. [S.l.], 2015. Citado na página 33.

INEP. Instituto nacional de estudos e pesquisas educacionais anísio teixeira. *Censo escolar da educação básica*, 2013. Citado na página 13.

KOLEN, M. J.; TONG, Y. Psychometric properties of irt proficiency estimates. *Educational Measurement: Issues and Practice*, Wiley Online Library, v. 29, n. 2, 2010. Citado na página 21.

KOULOURI, T.; LAURIA, S.; MACREDIE, R. D. Teaching introductory programming: A quantitative evaluation of different approaches. *ACM Transactions on Computing Education (TOCE)*, ACM New York, NY, United States, v. 14, n. 4, 2014. Citado na página 32.

LISHINSKI, A. et al. The influence of problem solving abilities on students performance on different assessment tasks in cs1. In: ACM MEMPHIS, TN, UNITED STATES. *Proceedings of the Special Interest Group on Computer Science Education (SIGCSE)*. [S.l.], 2016. Citado 3 vezes nas páginas 13, 17 e 32.

LU, I. R. R.; THOMAS, D. R.; ZUMBO, B. D. Embedding irt in structural equation models: A comparison with regression based on irt scores. *Structural Equation Modeling*, Taylor & Francis, v. 12, n. 2, 2005. Citado na página 21.

MALIK, S. I. Enhancing practice and achievement in introductory programming using an adri editor. In: IEEE, BANGKOK, THAILAND. *Proceedings of the IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. [S.l.], 2016. Citado na página 32.

MATHEW, R.; MALIK, S. I.; TAWAFK, R. M. Teaching problem solving skills using an educational game in a computer programming course. *Informatics in Education*, v. 18, n. 2, 2019. Citado na página 17.

MICHAELI, T.; ROMEIKE, R. Current status and perspectives of debugging in the k12 classroom: A qualitative study. In: IEEE, DUBAI, UNITED ARAB EMIRATES. *Proceedings of the IEEE Global Engineering Education Conference (EDUCON)*. [S.l.], 2019. Citado na página 16.

MULLER, O.; BUTMAN, A.; BUTMAN, M. Opening a (sliding) window to advanced topics. In: ACM BOLOGNA, ITALY. *Proceedings of the ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)*. [S.l.], 2017. Citado na página 17.

NUNNALLY, J. C. *Psychometric Theory 3E*. [S.l.]: Tata McGraw-Hill Education, 1994. Citado na página 19.

PARK, C. J.; HYUN, J. S. Effects of abstract thinking and familiarity with programming languages on computer programming ability in high schools. In: IEEE, WELLINGTON, NEW ZEALAND. *Proceedings of the International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. [S.l.], 2014. Citado 4 vezes nas páginas 13, 18, 34 e 35.

PARK, C. J.; HYUN, J. S.; HEUILAN, J. Effects of gender and abstract thinking factors on adolescents' computer program learning. In: IEEE, EL PASO, TX, UNITED STATES. *Proceedings of the Frontiers in Education Conference (FIE)*. [S.l.], 2015. Citado 3 vezes nas páginas 13, 18 e 34.

PASQUALI, L. *Psicometria: Teoria dos Testes na Psicologia e na Educação*. [S.l.]: Editora Vozes Limitada, 2017. Citado 3 vezes nas páginas 19, 21 e 22.

PRIMI, R. *Psicometria: Fundamentos matemáticos da teoria clássica dos testes*. *Avaliação Psicológica*, Instituto Brasileiro de Avaliação Psicológica, v. 11, n. 2, 2012. Citado na página 19.

ROSE, S. Bricolage programming and problem solving ability in young children: An exploratory study. In: UNIVERSITY OF THE WEST OF SCOTLAND, PAISLEY, SCOTLAND. *Proceedings of the European Conference on Games Based Learning*. [S.l.], 2016. Citado na página 17.

SOUZA, L. M. et al. Mathematics and programming: Marriage or divorce? In: IEEE, LIMA, PERU. *Proceedings of the World Conference on Engineering Education (EDUNINE)*. [S.l.], 2019. Citado 4 vezes nas páginas 13, 18, 19 e 35.

STATTER, D.; ARMONI, M. Teaching abstraction in computer science to 7th grade students. *ACM Transactions on Computing Education (TOCE)*, ACM New York, NY, United States, v. 20, n. 1, 2020. Citado na página 18.

TAHERI, S. M.; HIDEHIKO, Y.; TRIPATHY, H. K. Novel assessment of different intelligent tools for problem solving. *Computer Science and Engineering*, v. 3, n. 3, 2013. Citado 2 vezes nas páginas 17 e 33.

WATSON, C.; LI, F. W. Failure rates in introductory programming revisited. In: ACM. *Proceedings of the 2014 conference on Innovation & technology in computer science education*. [S.l.], 2014. p. 39–44. Citado na página 13.

ANEXO A – TERMO DE ASSENTIMENTO

Caro Responsável/Representante legal,

Convidamos o menor a participar do projeto de pesquisa “**Mensuração de habilidades cognitivas introdutórias de programação por meio de uma avaliação adaptativa informatizada**” coordenado pelos professores Dr. Wilkerson de Lucena Andrade e Dr. João Arthur Brunet Monteiro vinculados ao Programa de Pós Graduação da Universidade Federal de Campina Grande, a quem poderar contatar/consultar a qualquer momento que julgar necessário através dos e-mails {wilkerson, joao.arthur}@computacao.ufcg.edu.br. O documento abaixo contém todas as informações necessárias sobre a pesquisa que estamos fazendo. A colaboração do menor neste estudo será de muita importância para nós, mas se desistir a qualquer momento, isso não causará nenhum prejuízo a ele.

Eu,, portador da Cédula de identidade, RG, e inscrito no CPF....., abaixo assinado(a), concordo de livre e espontânea vontade em autorizar a participação do menor como voluntário(a) deste estudo. Declaro que obtive todas as informações necessárias, bem como todos os eventuais esclarecimentos quanto às dúvidas por mim apresentadas.

Estou ciente que:

- Este estudo tem por objetivo mensurar as habilidades cognitivas introdutórias relacionadas à programação de alunos em cursos técnicos e superiores com intuito de analisar suas limitações e descrever métodos para aprimorar o processo de ensino-aprendizagem.
- O menor será submetido aos seguintes procedimentos: i) fornecer respostas à exercícios durante a aplicação de instrumentos psicométricos; e, ii) avaliar o seu nível de satisfação ao usar os instrumentos psicométricos. Na qual serei beneficiado por avaliar minhas habilidades introdutórias em Programação.
- Caso o menor sinta riscos por está intimidado(a) ou desconfortado(a) durante a participação na pesquisa, poderá desistir a qualquer momento, retirando seu consentimento, sem que isso lhe traga nenhum prejuízo ou penalidade e receberá apoio necessário a questão apresentada.
- Todas as informações obtidas serão sigilosas e o nome do menor não será identificado em nenhum momento. Os dados serão guardados em local seguro e a divulgação dos resultados será feita de maneira que não permita a identificação dele.
- Se o menor tiver algum gasto decorrente a participação na pesquisa, será ressarcido, caso solicite. Em qualquer momento, se o menor sofrer algum dano comprovadamente decorrente desta pesquisa, será indenizado.
- Caso o menor se sinta prejudicado (a) por participar desta pesquisa, poderei recorrer ao Comitê de Ética em Pesquisas com Seres Humanos – CEP, do Hospital Universitário Alcides Carneiro - HUAC, situado a Rua: Dr. Carlos Chagas, s/ n, São José, CEP: 58401 – 490, Campina Grande-PB, Tel: 2101 – 5545, E-mail: cep@huac.ufcg.edu.br; Conselho Regional de Medicina da Paraíba e a Delegacia Regional de Campina Grande.
- Atesto recebimento de uma via assinada deste Termo de Assentimento, conforme recomendações da Comissão Nacional de Ética em Pesquisa (CONEP). Outros esclarecimentos sobre esta pesquisa, poderei entrar em contato com o pesquisador principal Me. Jucelio Soares dos Santos, Rua Severino Pimentel através do e-mail jucelio@copin.ufcg.edu.br.

Campina Grande - PB, de de 20

(Assinatura do responsável ou representante legal)

(Testemunha 1 | Nome/RG/Telefone)

(Testemunha 2 | Nome/RG/Telefone)

**Me. Jucelio Soares dos Santos
(Responsável pelo projeto)**

ANEXO B – TERMO DE CONSENTIMENTO LIVRE ESCLARECIDO

Você está sendo convidado(a) a participar do projeto de pesquisa “**Mensuração de habilidades cognitivas introdutórias de programação por meio de uma avaliação adaptativa informatizada**” coordenado pelos professores Dr. Wilkerson de Lucena Andrade e Dr. João Arthur Brunet Monteiro vinculados ao Programa de Pós Graduação da Universidade Federal de Campina Grande, a quem poderar contactar/consultar a qualquer momento que julgar necessário através dos e-mails {wilkerson, joao.arthur}@computacao.ufcg.edu.br. O documento abaixo contém todas as informações necessárias sobre a pesquisa que estamos fazendo. Sua colaboração neste estudo será de muita importância para nós, mas se desistir a qualquer momento, isso não causará nenhum prejuízo a você.

Eu,, portador da Cédula de identidade, RG, e inscrito no CPF....., abaixo assinado(a), concordo de livre e espontânea vontade em participar como voluntário(a) deste estudo. Declaro que obtive todas as informações necessárias, bem como todos os eventuais esclarecimentos quanto às dúvidas por mim apresentadas.

Estou ciente que:

- Este estudo tem por objetivo mensurar as habilidades cognitivas introdutórias relacionadas à programação de alunos em cursos técnicos e superiores com intuito de analisar suas limitações e descrever métodos para aprimorar o processo de ensino-aprendizagem.
- Serei submetido aos seguintes procedimentos: i) fornecer respostas à exercícios durante a aplicação de instrumentos psicométricos; e, ii) avaliar o seu nível de satisfação ao usar os instrumentos psicométricos. Na qual serei beneficiado por avaliar minhas habilidades introdutórias em Programação.
- Caso sinta riscos por está intimidado(a) ou desconfortado(a) durante a participação na pesquisa, poderei desistir a qualquer momento, retirando meu consentimento, sem que isso me traga nenhum prejuízo ou penalidade e receberei apoio necessário a questão apresentada.
- Todas as informações obtidas serão sigilosas e meu nome não será identificado em nenhum momento. Os dados serão guardados em local seguro e a divulgação dos resultados será feita de maneira que não permita a minha identificação.
- Se eu tiver algum gasto decorrente de minha participação na pesquisa, serei ressarcido, caso solicite. Em qualquer momento, se eu sofrer algum dano comprovadamente decorrente desta pesquisa, serei indenizado.
- Caso me sinta prejudicado (a) por participar desta pesquisa, poderei recorrer ao Comitê de Ética em Pesquisas com Seres Humanos – CEP, do Hospital Universitário Alcides Carneiro - HUAC, situado a Rua: Dr. Carlos Chagas, s/ n, São José, CEP: 58401 – 490, Campina Grande-PB, Tel: 2101 – 5545, E-mail: cep@huac.ufcg.edu.br; Conselho Regional de Medicina da Paraíba e a Delegacia Regional de Campina Grande.
- Atesto recebimento de uma via assinada deste Termo de Consentimento Livre e Esclarecido, conforme recomendações da Comissão Nacional de Ética em Pesquisa (CONEP). Outros esclarecimentos sobre esta pesquisa, poderei entrar em contato com o pesquisador principal Me. Jucelio Soares dos Santos através do e-mail: jucelio@copin.ufcg.edu.br.

Campina Grande - PB, de de 20

(Assinatura do participante)

(Testemunha 1 | Nome/RG/Telefone)

(Testemunha 2 | Nome/RG/Telefone)

Me. Jucelio Soares dos Santos
(Responsável pelo projeto)

APÊNDICE A – INSTRUMENTO DE APTIDÃO EM PROGRAMAÇÃO



INSTRUMENTO DE APTIDÃO EM PROGRAMAÇÃO

CURSO	Ciência da Computação	
NOME		
MATRÍCULA		CONCEITO
DATA		

1. Crie um programa que receba três notas de um aluno (em float) e uma letra. Se a letra for 'A', a calcule a média aritmética das notas do aluno; se a letra for 'P', deverá calcular a média ponderada, com pesos 5, 3 e 2. Retorne a média calculada.

(A média ponderada é calculada por meio da multiplicação de cada nota pelos pesos 5, 3 e 2, respectivamente e dividir pela soma dos pesos).

2. Escreva um programa para somar todos os dígitos de uma entrada com 3 dígitos. (Ex: 320 ⇒ 3 + 2 + 0 = 5)

Para resolver essa questão é necessário adotar o processamento de dígitos. A seguir, apresentamos o pseudocódigo de como processar dígitos.

1. Comece com um número inteiro como entrada;

2. Repita as etapas a seguir até que todos os dígitos tenham sido processados:

a) Use o operador de módulo para obter o último dígito e armazenar esse valor;

b) Use o operador de divisão para remover o último dígito.

3. Considere o seguinte trecho de código em Python. Assinale a alternativa em que este comando pode ser escrito de forma mais simples como:

```
if p==True:
    print(True)
elif q==True:
    print(True)
else:
    print(False)
```

a.
if p==False and q==True:
 print(True)
else:
 print(False)

b.
if p==True and q==False:
 print(True)
else:
 print(False)

c.
if p==True or q==True:
 print(True)
else:
 print(False)

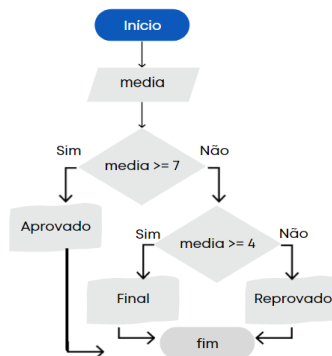
d.
if p==False or q==False:
 print(True)
else:
 print(False)

4. Crie um programa que imprima a soma do maior e menor valor entre as variáveis x, y e z.

(Ex: Entre x = 3, y = 4 e z = 5, retorna maior + menor = 5 + 3 = 8)

5. Construa um programa que receba a idade de uma pessoa e identifique sua classe eleitoral: não eleitor (menor que 16 anos de idade), eleitor obrigatório (entre 18 e 65 anos) e eleitor facultativo (entre 16 e 18 anos e maior que 65 anos).

6. Interprete o fluxograma abaixo, verifique a alternativa onde traduz a passagem de código para uma instrução de seleção aninhada para Python.



a)
if (media>=7):
 print("Aprovado")
elif (media>=4):
 print("Final")
else:
 print("Reprovado")

b)
if (media>=7):
 print("Aprovado")
else if (media>=4):
 print("Final")
else:
 print("Reprovado")

c)
if (media>=7):
 print("Aprovado")
if else (media>=4):
 print("Final")
else:
 print("Reprovado")

d)
if (media>=7):
 print("Aprovado")
else (media>=4):
 print("Final")
elif:
 print("Reprovado")

7. Crie uma função que receba a altura (em metros) e o peso (em quilos) e verifique o IMC e seu status conforme a tabela abaixo.

O IMC é calculado por $\text{peso} / \text{altura}^2$

Intervalo	Status
IMC abaixo de 20	Abaixo do Peso
IMC entre 20 e 25	Peso ideal
IMC acima de 25	Acima do Peso

8. A função abaixo está incompleta. Esta função verifica se um número é primo. Preencha os espaços vazios e assinale a alternativa correta. Um número primo é um número que tem apenas dois divisores que são 1 e ele mesmo.

<pre>#Função primo def primo(n): count = 0 for i in range (__,n): if(n__i==0): count = count + 1 if(count==2): return 1 else: return 0</pre>	<pre>#Programa Principal n = int(input("Digite um inteiro: ")) result = primo(n) if result == __: print(n, " é primo") else: print(n, " não é primo")</pre>
--	---

- a. 0, %, 0
- b. 0, /, 1
- c. 1, /, 0
- d. 1, %, 1