



**UNIVERSIDADE ESTADUAL DA PARAÍBA
CAMPUS I - CAMPINA GRANDE
CENTRO DE CIÊNCIAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO
CURSO DE GRADUAÇÃO EM BACHARELADO EM COMPUTAÇÃO**

PEDRO HENRIQUE DE ANDRADE

**ANÁLISE DA USABILIDADE DE UM SISTEMA WEB NA MENSURAÇÃO DE
HABILIDADES COGNITIVAS PREDITORAS EM NOVATOS EM PROGRAMAÇÃO**

CAMPINA GRANDE - PB

2022

PEDRO HENRIQUE DE ANDRADE

**ANÁLISE DA USABILIDADE DE UM SISTEMA WEB NA MENSURAÇÃO DE
HABILIDADES COGNITIVAS PREDITORAS EM NOVATOS EM PROGRAMAÇÃO**

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

Orientador: Dra. Kézia de Vasconcelos Oliveira Dantas

CAMPINA GRANDE - 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.

A553a Andrade, Pedro Henrique de.
Análise da usabilidade de um sistema web na mensuração de habilidades cognitivas preditoras em novatos em programação [manuscrito] / Pedro Henrique de Andrade. - 2022.
58 p. : il. colorido.

Digitado.
Trabalho de Conclusão de Curso (Graduação em Computação) - Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia, 2022.
"Orientação : Profa. Dra. Kézia de Vasconcelos Oliveira Dantas, Coordenação do Curso de Computação - CCT."
1. Programação de computadores. 2. Sistema Web. 3. Usabilidade. I. Título

21. ed. CDD 005.3

PEDRO HENRIQUE DE ANDRADE

ANÁLISE DA USABILIDADE DE UM SISTEMA WEB NA MENSURAÇÃO DE
HABILIDADES COGNITIVAS PREDITORAS EM NOVATOS EM PROGRAMAÇÃO

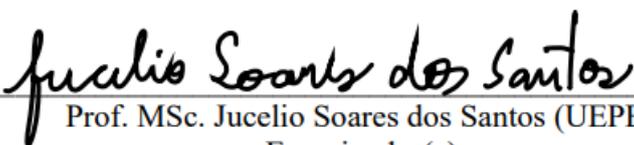
Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Computação do Departamento de Computação do Centro de Ciências e Tecnologia da Universidade Estadual da Paraíba, como requisito parcial à obtenção do título de bacharel em Computação.

Trabalho aprovado em 23 de Março de 2022.

BANCA EXAMINADORA


Prof. Dra. Kézia de Vasconcelos Oliveira Dantas (DC - UEPB)
Orientador(a)


Prof. Dr. Wellington Candeia de Araujo (DC - UEPB)
Examinador(a)


Prof. MSc. Jucelio Soares dos Santos (UEPB)
Examinador(a)

Dedico o presente trabalho primeiramente a Deus, por ter me mantido forte e paciente ao decorrer desta jornada. Aos meus pais Abílio Gomes Cavalcante e Maria Lucimar de Andrade, que me deram todo o impulso necessário. Aos meus amigos, que também me ajudaram nessa longa caminhada.

AGRADECIMENTOS

Primeiramente a Deus, por me ajudar a superar todas as dificuldades pelas quais eu passei.

Aos meus familiares, por terem me dado todo o apoio e incentivo necessário.

A minha orientadora, Prof. Dra. Kézia de Vasconcelos Oliveira Dantas pela contribuição e apoio para a realização deste trabalho. Agradeço pelo esforço e tempo dedicados nas correções

Ao Prof. Me. Jucelio Soares dos Santos pela contribuição e apoio para a realização deste trabalho.

Agradeço a SPLAB, a Cognite, a CNPq, ao departamento de computação da UEPB e ao departamento de computação da UFCG pelas condições de logística para realização deste estudo.

Aos meus amigos Wylliam Eduardo e Gilmar Barbosa, que conviveram comigo desde o início do curso aqui em Campina Grande.

Aos meus amigos de Bacharelado de computação, com os quais convivi durante toda a minha jornada de curso.

Aos professores e funcionários da UEPB, que fizeram parte de toda a minha trajetória.

*“Devemos aprender durante toda a vida,
sem imaginar que a sabedoria vem com a velhice.”*

Platão

RESUMO

Este trabalho promove reflexões sobre ferramentas para identificar sinais de falhas em ingressantes de cursos Técnicos e Superiores nas disciplinas Introdutórias de Programação (CS1). O intuito é sistematizar um teste feito em papel-e-lápis e, analisar a sua usabilidade durante o processo de mensuração das habilidades cognitivas preditoras em CS1. Para tanto, concebemos/desenvolvemos o Screening Programming com base em um banco de itens calibrados; elaboramos e aplicamos um Survey aos estudantes em um CS1 com o objetivo de avaliar dimensões da usabilidade com base nos comentários e observações feitas pelos usuários em relação a sua aplicação, a fim de determinar as vantagens e limitações. A principal contribuição desta pesquisa foi a sistematização de um instrumento em papel-e-lápis e análise da sua usabilidade, permitindo assim, o desenvolvimento de um produto agradável a seus usuários e que possa avaliar as habilidades preditoras de programação de uma forma mais confiável. Os resultados apontaram que o sistema apresenta uma diferença significativa entre os usuários que se agradaram durante a sua aplicação, como apontam os dados para as métricas: “utilidade”, “facilidade de usar”, “facilidade de aprender” e “satisfação”. O que dá fortes indicativos que os usuários obtiveram uma empatia pelo rastreamento/evolução das habilidades preditoras de programação feito por meio de uma versão informatizada. Ressaltamos ainda, que os resultados alcançados nesta pesquisa se aplicam apenas para a realidade dos usuários da área em estudo e não podem ser generalizados.

Palavras-chaves: Sistema Web. Triagem. Habilidades Preditoras de Programação. Usabilidade.

ABSTRACT

This work promotes reflections on tools to identify signs of failure in newcomers of Technical and Higher courses in the Introductory Programming disciplines (CS1). The aim is to systematize a paper-and-pencil test and analyze its usability while measuring predictive cognitive skills in CS1. To this end, we conceived/developed Screening Programming based on an item bank calibrated; We designed and applied a Survey to students in a CS1 to evaluate usability dimensions based on users' comments and observations regarding its application to determine the advantages and limitations. This research's main contribution was the instrument's systematization in paper-and-pencil and analysis of its usability, thus allowing the development of a pleasant product to its users that can evaluate the predictive programming skills more reliably. The results showed that the system presents a significant difference between users who were pleased during its application, as shown by the data for the metrics: utility, ease of use, ease of learning, and satisfaction. These results give strong indications that users have gained empathy for tracking/evolving predictive programming skills done through a computerized version. We also emphasize that the results achieved in this research apply only to the users' reality in the area under study and cannot be generalized.

Keywords: Web System. screening. Predictive Programming Skills. Usability.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama de Caso de Uso dos Usuários Visitante e Administrador	35
Figura 2 – Diagrama de Caso de Uso dos Usuários Aluno e Professor	35
Figura 3 – Arquitetura do <i>Screening Programming</i>	37
Figura 4 – Página inicial do <i>Screening Programming</i>	38
Figura 5 – Página inicial do <i>Screening Programming</i>	38
Figura 6 – Página com informações sobre o <i>Screening Programming</i>	39
Figura 7 – Página de créditos do <i>Screening Programming</i>	39
Figura 8 – Página de contatos.	40
Figura 9 – Página de cadastro de usuário.	40
Figura 10 – Página de continuar cadastro do Professor.	41
Figura 11 – Página inicial do Professor com informações dos grupos criados.	41
Figura 12 – Página de cadastro de grupos.	42
Figura 13 – Página de convite dos alunos aos grupos.	42
Figura 14 – Página de acesso ao grupo.	43
Figura 15 – Página de acesso às estatísticas.	43
Figura 16 – Página para acessar cada instrumento avaliativo.	44
Figura 17 – Página de continuar cadastro do Aluno.	44
Figura 18 – Página inicial do Aluno com informações dos instrumentos disponíveis.	45
Figura 19 – Página de aplicação do instrumento avaliativo.	45
Figura 20 – Página do desempenho final do Aluno.	46

LISTA DE TABELAS

Tabela 1 – Análise dos resultados sobre a opinião dos respondentes para Utilidade . . .	47
Tabela 2 – Análise dos resultados sobre a opinião dos respondentes para Facilidade de Usar	48
Tabela 3 – Análise dos resultados sobre a opinião dos respondentes para Facilidade de Aprender	49
Tabela 4 – Análise dos resultados sobre a opinião dos respondentes para Satisfação . .	49

LISTA DE QUADROS

Quadro 1 – Problema da soma de três preços de supermercado	18
Quadro 2 – Algoritmo de solução do problema da soma de três preços de supermercado	19
Quadro 3 – Problema da soma de três preços gerais	19
Quadro 4 – Indicadores de dimensão do modelo USE	28

LISTA DE ABREVIATURAS E SIGLAS

CS1	Cursos Introdutórios de Programação
SPLab	Laboratório de Práticas de Software
UEPB	Universidade Estadual da Paraíba
UFCG	Universidade Federal de Campina Grande

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Cenário Técnico Científico	14
1.2	Problema e Proposta de Solução	15
1.3	Objetivos	16
1.4	Questões de Pesquisa	16
1.5	Estrutura do Trabalho	17
2	TEMAS E TRABALHOS RELACIONADOS	18
2.1	Metacognição e Programação	18
2.2	Habilidade Predictoras de Programação	20
2.2.1	<i>Resolução de Problemas</i>	20
2.2.2	<i>Pensamento Abstrato</i>	21
2.2.3	<i>Raciocínio Matemático</i>	21
2.2.4	<i>Flexibilidade Cognitiva</i>	22
2.3	Usabilidade	23
3	METODOLOGIA	24
3.1	<i>Screening Programming</i>	24
3.1.1	<i>Análise de Planejamento</i>	24
3.1.2	<i>Design</i>	25
3.1.3	<i>Desenvolvimento</i>	26
3.1.4	<i>Validação e Arranque</i>	27
3.2	Elaboração do Questionário	27
3.3	Aplicação do Questionário	28
3.3.1	<i>Preparação</i>	29
3.3.2	<i>Análise dos Dados</i>	29
3.3.3	<i>Análise às ameaças</i>	29
3.3.4	<i>Execução da Pesquisa</i>	30
3.3.5	<i>Divulgação</i>	30
4	SCREENING PROGRAMMING	31
4.1	Análise de Requisitos	31
4.1.1	<i>Requisitos Funcionais</i>	31
4.1.2	<i>Requisitos Não-funcionais</i>	33
4.2	Modelo de Análise	33
4.2.1	<i>Perfil do Usuário</i>	33
4.2.2	<i>Sequência de Eventos do Instrumento</i>	34
4.2.3	<i>Diagrama de Caso de Uso</i>	34

4.2.4	<i>Tecnologias de Suporte</i>	36
4.3	Modelo de Arquitetura	36
4.4	Modelo de Projeto	37
4.4.1	<i>Pré-autenticação</i>	37
4.4.2	<i>Pós-autenticação Professor</i>	40
4.4.3	<i>Pós-autenticação Aluno</i>	44
5	ANÁLISE E RESULTADOS	47
5.1	Análise da Utilidade	47
5.2	Análise da Facilidade de Usar	48
5.3	Análise da Facilidade de Aprender	48
5.4	Análise de Satisfação	49
6	CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS	51
	REFERÊNCIAS	52
	ANEXO A – TERMO DE ASSENTIMENTO	57
	ANEXO B – TERMO DE CONSENTIMENTO LIVRE ESCLARECIDO	58

1 INTRODUÇÃO

Neste capítulo, apresentamos uma visão geral desta pesquisa, de modo a descrever o cenário técnico científico, problema, objetivos e questões de pesquisas.

1.1 Cenário Técnico Científico

Os Cursos Introdutórios de Programação (CS1) apresentam altas taxas de evasão e reprovação, correspondendo a um terço dos alunos matriculados nessa disciplina mundialmente (WATSON; LI, 2014). Os alunos que iniciam um CS1 possuem diferentes habilidades prévias, impossibilitando o professor de criar um processo que atenda a diferentes perfis.

Nesse cenário, o professor não tem escolha: ou intensifica o ritmo das aulas e a complexidade para incentivar os alunos mais avançados, ou minimiza o ritmo e a complexidade para permitir que os alunos com dificuldades possam acompanhar a disciplina. Mesmo conduzindo em um ritmo mediano, como forma de beneficiar a todos, os alunos deixam de conceber a grande diversidade de pensamentos, compreensões, ritmos e estilos de aprendizagem que os diferentes perfis de estudantes possuem. Independente da escolha, uma parcela da turma acabará sendo prejudicada. Ao utilizarmos a abordagem tradicional, falhamos em dar apoio suficiente a todos os estudantes nas atividades que envolvem a aprendizagem e o reforço e, no processo de avaliação (MORAIS, 2015).

Similarmente, as instituições de Ensino Superior usam estratégias para contornar e nivelar o conhecimento dos estudantes em matemática, por exemplo. As instituições usam estratégias para avaliar a habilidade prévia dos alunos em matemática básica, tentando desenvolver intervenções nesses acadêmicos para evitar falhas e evasão em disciplinas que envolvem habilidade, em algum nível, de cálculo matemático (MEDEIROS, 2019). Quando levamos essa mesma realidade do problema para o cenário do ensino de programação nas instituições de Ensino Superior, partimos do pressuposto que estratégias como essas possam avaliar as habilidades precursoras à programação em ingressantes do curso de Ciência da Computação e; este por sua vez, seja um forte aliado quando partimos da perspectiva de que os iniciantes vêm de instituições diferentes, tiveram estímulos diferentes do desenvolvimento de habilidades antecessoras à programação, entre outros fatores. Dessa forma permitirá ao professor mensurar/estimular o nível de habilidades de seus alunos para as disciplinas do primeiro semestre, de tal forma que ministraria suas aulas cientes das habilidades preditoras¹ que os alunos têm maior domínio ou não.

Os educadores de informática devem ajustar o *design* instrucional do Ensino Tradicional (CABO; LANSQUOT, 2014) e, estimar as habilidades de programação dos alunos com antecedência. As habilidades discutidas na literatura são **resolução de problemas**² (ALSHAYE; TASIR; JUMAAT, 2019) (BASNET et al., 2018) (CABO; LANSQUOT, 2014) (CHAO, 2016)

¹ Habilidades Preditoras são habilidades específicas que antecedem um marco ou o desenvolvimento de outras habilidades mais estruturadas, neste caso, habilidades que antecedem as habilidades de programação.

² Capacidade do aluno em conseguir encontrar uma solução para resolver um determinado problema;

(CHAUDHRY; RASOOL, 2012) (CHETTY; BARLOW-JONES, 2014) (ÇOKLAR; AKCAY, 2018) (DEKHANE; XU; TSOI, 2019) (DURAK, 2018) (DURAK, 2020) (FIGUEIREDO; GARCIA-PENALVO, 2017) (FIGUEIREDO; GARCÍA-PEÑALVO, 2019) (FIGUEIREDO; NATÁLIA; GARCÍA-PEÑALVO, 2016) (GOMES et al., 2017) (HOOSHYAR et al., 2014) (HOOSHYAR; AHMAD; NASIR, 2014) (HOOSHYAR et al., 2015) (HOOSHYAR et al., 2015a) (HOOSHYAR et al., 2015b) (HUEI, 2014) (JAKOŠ; VERBER, 2017) (JONES; WESTHUIZEN, 2017) (KOULOURI; LAURIA; MACREDIE, 2014) (LANSIQUOT; CABO, 2015) (LISHINSKI et al., 2016) (MALIK, 2016) (MALIK; COLDWELL-NEILSON, 2017) (MATHEW; MALIK; TAWAFK, 2019) (MLADENOVIC; KR PAN; MLADENOVIC, 2017) (MULLER; BUTMAN; BUTMAN, 2017) (NIEMELÄ et al., 2017) (PALMER; FLIEGER; HILLENBRAND, 2011) (PAPADOPOULOS; TEGOS, 2012) (PHILIP; RENUMOL; GOPEEKRISHNAN, 2013) (ROSE, 2016) (SANDS, 2019) (SKALKKA; DRLÍK, 2018) (SPANGSBERG; FINCHER; DZIALLAS, 2018) (TAHERI; HIDEHIKO; TRIPATHY, 2013) (THEVATHAYAN; HAMILTON, 2015) (TOPALLI; CAGILTAY, 2018) (VEERASAMY et al., 2019), **pensamento abstrato**³ (CHETTY; BARLOW-JONES, 2014) (LÓPEZ et al., 2017) (PARK; HYUN, 2014) (PARK; HYUN; HEUILAN, 2015) (PHILIP; RENUMOL; GOPEEKRISHNAN, 2013) (SKALKKA; DRLÍK, 2018) (STATTER; ARMONI, 2020), **raciocínio matemático**⁴ (ATTALLAH; ILAGURE; CHANG, 2018) (DRACHOVA et al., 2015) (GOMES et al., 2017) (SOUZA et al., 2019) e **flexibilidade cognitiva**⁵ (DURAK, 2018) (DURAK, 2020).

1.2 Problema e Proposta de Solução

Apesar dos avanços nos métodos, processos, abordagens e instrumentos para o ensino e aprendizagem em CS1, as taxas de evasão e reprovação ainda são altas (SANDS, 2019). Na prática, os aspectos para identificar as limitações em um novato em programação torna-se uma problemática para os educadores em CS1. O processo de identificação do nível de desenvolvimento de indivíduos em programação pelos professores é criado de forma habitual a papel-e-lápis, o que pode causar uma certa demora para concluir o resultado do diagnóstico da evolução no cotidiano atual.

Diante do exposto problema enfrentado por esses profissionais, em parceria com o Laboratório de Neuropsicologia Cognitiva e Inovação Tecnológica da Universidade Federal de Campina Grande (UFCG), o Laboratório de Práticas de Software (SPLab) e a Universidade Estadual da Paraíba (UEPB) foi desenvolvido um sistema web destinado a rastrear habilidades predictoras em programação. Acreditando que uma forma de contornar essas limitações é propor a instrumentalização desse processo na criação de aplicações que proporcione um apoio significativo levando em consideração uma série de requisitos que atenda às especificidades dentro do

³ Capacidade do aluno em conseguir dividir um determinado problema em partes menores e resolvê-los separadamente;

⁴ Capacidade do aluno em ter facilidade com matemática e isso o ajuda a resolver um problema;

⁵ Capacidade do aluno em pensar em diferentes soluções para o mesmo problema.

CS1.

Os cenários existentes de produção de sistemas com qualidade para o mercado ampliam o nível de condições expostas pelos usuários. A concorrência realiza técnicas e métodos para tornar o produto otimizado. Melhorar a agilidade, usabilidade e contenção de riscos e falhas são condições que podem interferir na aceitação ou não do produto. Nessas circunstâncias, o presente trabalho não se limitou apenas em desenvolver um sistema web, mas também teve como finalidade analisar a sua usabilidade. Se tratando de sistemas com finalidade de rastrear as habilidades predictoras de programação, os profissionais enfatizam que sentem uma grande dificuldade no mercado de instrumentos que façam esse tipo de avaliação. Diante dessa escassez surgiu a necessidade da implementação do *Screening Programming*.

1.3 Objetivos

A presente pesquisa teve como objetivo principal sistematizar um teste feito em papel-e-lápis e, analisar a sua usabilidade durante o processo de mensuração das habilidades cognitivas predictoras em novatos em programação⁶.

Para alcançar o objetivo geral desta pesquisa, foram necessários atingir os seguintes objetivos específicos:

- Sistematizar um teste em papel-e-lápis para avaliações das habilidades cognitivas predictoras em novatos em programação;
- Elaborar e aplicar um questionário aos usuários em um CS1 com o objetivo de avaliar dimensões com base na usabilidade, em relação às métricas: utilidade, facilidade de uso, facilidade de aprendizado e satisfação;
- Analisar o sistema desenvolvido, com base nos comentários e observações feitas pelos aplicadores em relação a usabilidade, a fim de determinar as vantagens e limitações.

1.4 Questões de Pesquisa

A principal questão de pesquisa deste estudo é: O instrumento feito, a partir do teste em papel-e-lápis, provê uma usabilidade melhor no rastreamento das habilidades predictoras de programação? A fim de entender melhor se a aplicação possui uma melhor usabilidade, esta pesquisa se concentra nas seguintes Questões de Pesquisa (QP) dentro do escopo mais amplo da avaliação de habilidades prévias de programação:

- **QP1.** Qual metodologia de desenvolvimento é utilizada para instrumentalizar um teste em papel-e-lápis?

⁶ Ressaltamos que esse Trabalho de Conclusão de Curso foi fruto de parceria de uma pesquisa doutoral desenvolvida na UFCG, junto ao Programa de Pós-Graduação em Ciência da Computação, com o Prof. Me. Jucelio Soares dos Santos e seus orientadores Prof. Dr. Wilkerson de Lucena Andrade e Prof. João Arthur Brunet Monteiro como autor da tese.

- **QP2.** Quais métricas são utilizadas para avaliar usabilidade?
- **QP3.** O sistema desenvolvido apresenta uma boa usabilidade?

1.5 Estrutura do Trabalho

Este trabalho apresenta cinco capítulos e está organizado da seguinte maneira: no Capítulo 1, apresentamos uma visão geral desta investigação com relação ao cenário técnico científico, problema, objetivos e questões de pesquisa; no Capítulo 2, apresentamos os temas e trabalhos relacionados à pesquisa; no Capítulo 3, apresentamos a metodologia desta pesquisa; no Capítulo 4, apresentamos o *Screening Programming*; no Capítulo 5, apresentamos e discutimos os resultados; no Capítulo 6, apresentamos 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, apresentamos o embasamento teórico a partir de várias áreas e trabalhos que se relacionam e que permitem caracterizar esta pesquisa.

2.1 Metacognição e Programação

Imagine como seria nossa vida se não tivéssemos consciência de nossos pensamentos. Como podemos planejar nossas ações e corrigi-las quando não saem como o esperado? Como podemos monitorar nossos comportamentos e adaptá-los a cada necessidade que enfrentamos? Como podemos escolher a maneira mais adequada de estudar durante toda a vida?

Podemos fazer tudo isso a qualquer momento devido ao nosso pensamento de pensar sobre nós mesmos. A compreensão das pessoas sobre seu processamento cognitivo é chamada de metacognição (RHODES, 2019). As habilidades cognitivas e metacognitivas são assimiladas pelo ser humano desde o nascimento e se desenvolvem ao longo da vida. As habilidades cognitivas estão relacionadas às funções psicológicas superiores, sendo em si a capacidade de absorver conhecimento. Os estímulos naturais que recebemos desde o primeiro momento de nossas vidas nos ajudam a entender, aos poucos, como as coisas funcionam, mesmo que não seja diretamente ensinado sobre isso. Nesta subseção, apresentaremos como a metacognição pode ajudar os iniciantes a programar melhor.

Programar não é digitar códigos (LISHINSKI et al., 2016). Envolve resolver um problema usando lógica e só então codificar essa solução. Se um aluno não encontrar uma solução para um problema, ele também não a programará. Para entender melhor, fornecemos no Quadro 1 um exemplo de um problema de cálculo.

Quadro 1 – Problema da soma de três preços de supermercado

Suponha que uma pessoa foi a um supermercado e comprou três itens pelos seguintes preços: R\$7,54, R\$8,56 e R\$3,53. Calcule mentalmente o preço dos produtos.

Solução: Resultado é R\$19,63.

Fonte: Elaborada pelo autor (2022).

O mais importante é o caminho mental que o aluno percorreu para chegar a essa solução. Existem várias maneiras mentais de realizar uma soma. Alguns deles, o aluno deve ter usado para chegar à solução correta. Por exemplo, o aluno pode usar a lógica de dividir o problema em partes (ver QUA. 2). O mais importante é pensar da maneira que o pensamento leva enquanto estamos resolvendo um problema. Também é importante observar que esse problema pode ser resolvido de outras maneiras.

Quadro 2 – Algoritmo de solução do problema da soma de três preços de supermercado

Passo 1: Considere apenas a parte fundamental dos preços, ignorando tudo após a vírgula. Some esses valores, adotando a lógica de somar valores mais fáceis de fazer, como somar 7 com 3 que são fáceis de somar mentalmente e somar com 8. Esse resultado parcial, 18, o aluno consegue guardar mentalmente;

Passo 2: Considere a parte fracionária dos preços (ou seja, após a vírgula) e desconsidere os números à direita. Assim, é possível somar com mais facilidade, fingindo que cada valor é de 50 centavos. Assim, totalizando R\$1,50. Em seguida, esse valor é adicionado ao valor anterior, totalizando R\$19,50;

Passo 3: Some os centavos, adotando a mesma lógica de antes, opere os números que são mais fáceis de somar mentalmente, por exemplo, 6 com 4 e depois some 3 totalizando 13 centavos. Além disso, acrescenta esse valor a R\$19,50, totalizando R\$19,63.

Fonte: Elaborada pelo autor (2022).

Em seguida no Quadro 3, apresentamos outro exercício:

Quadro 3 – Problema da soma de três preços gerais

Com um lápis e papel nas mãos, pense em um problema semelhante. Observe três preços e, em seguida, some-os mentalmente. Ao adicionar, preste atenção aos processos mentais adotados para chegar à solução. Em seguida, peça ao aluno para somar esses números novamente, usando papel e lápis e a lógica que aprendeu na escola. Além disso, deve-se observar as etapas que ele deu para chegar à solução.

Fonte: Elaborada pelo autor (2022).

O aluno provavelmente adotou duas maneiras diferentes de resolver o mesmo problema. Essa prática é a primeira etapa da solução de problemas, pois o programador pode resolver um problema de maneira diferente. Além disso, o aluno dividiu o problema em partes. Também usou a abstração para resolver esse problema quando ignorou detalhes específicos do problema e fingiu que era mais simples. A abstração seria eliminar os detalhes temporariamente para ajudar a resolver o problema.

Dividir um problema em partes, ignorando detalhes temporariamente para resolver um problema, são habilidades essenciais em Programação de Computadores (DURAK, 2020) (LISHINSKI et al., 2016) (PARK; HYUN, 2014) (SOUZA et al., 2019). Cada problema o programador resolve em partes menores e, à medida que são resolvidos, o programa como um todo vai acontecendo.

Resolver problemas inclui encontrar uma solução para resolver um problema específico. Alguns problemas podem ser mais complexos, exigindo que o aluno seja abstrato, divida-os em partes menores e resolva-os separadamente. Mesmo com esse domínio, é necessário ter facilidade com a matemática, pois ajuda a resolver um problema de forma mais eficiente. Finalmente, cada

problema tem várias maneiras de resolvê-lo, então o iniciante precisa encontrar outra solução para o mesmo problema de uma forma cognitivamente flexível.

2.2 Habilidade Predictoras de Programação

As habilidades predictoras de programação são resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva. Promoveremos nesta seção, a definição destas habilidades, bem como as principais estratégias para estimá-las.

2.2.1 Resolução de Problemas

A capacidade de observar, analisar e compreender um determinado problema é essencial em nosso dia a dia (CHAUDHRY; RASOOL, 2012). A busca pela melhor solução para um problema específico depende de experiências individuais, nível de proficiência e metacognição (LISHINSKI et al., 2016). Esses recursos podem ser aproveitados para facilitar a solução de problemas com maior eficiência.

A matemática é um campo que tem influência significativa em todas as esferas da computação, portanto, em CS1 não seria diferente (SOUZA et al., 2019). Um dos principais fatores na importância da resolução de problemas de programação é sua correlação com a matemática. Um dos déficits encontrados em alunos iniciantes em programação é a solução de problemas e intensificada pela sintaxe da linguagem. Esses fatores contribuem para essa situação (MULLER; BUTMAN; BUTMAN, 2017) (NIEMELÄ et al., 2017). Essa habilidade é essencial na programação, pois se um programador tiver maior capacidade de inferência e souber definir mentalmente formas de se chegar a uma solução específica para um problema, ele aplicará esses conhecimentos e terá sucesso em suas codificações.

Existem várias ferramentas e linguagens de programação simples disponíveis que ajudam nesta fase inicial de assimilação. Essas ferramentas, disponíveis principalmente online ou por meio de linguagens simples, são oferecidas na forma de competição por meio de desafios nos quais o programador iniciante pode estimular sua capacidade de resolver problemas e estabelecer direções para atingir os objetivos propostos. A linguagem de programação, em conjunto com a Integrated Development Environment, por sua vez, oferece uma gama de mecanismos e uma metodologia pré-definida em que o programador inexperiente pode implementar essas direções.

Alguns softwares servem como ferramentas educacionais para crianças e alunos iniciantes em programação. Um exemplo menos recente para esse propósito é uma linguagem de programação interpretada que surgiu do pensamento construtivista de Piaget. Esse pensamento é representado na linguagem do Logo pela figura da tartaruga que desenha linhas por meio do comando do usuário e na tela. Outra perspectiva são as linguagens Scratch e Tynker, por exemplo, que são plataformas educacionais que trabalham por meio de blocos de código que os alunos arrastam e ajustam de acordo com o objetivo da atividade.

Entre outras alternativas para auxiliar esses alunos em suas habilidades de resolução de

problemas, está o Bebras (SPANGSBERG; FINCHER; DZIALLAS, 2018), um instrumento com um sistema de desafios, tarefas e competições para colaborar no desenvolvimento da capacidade de pensamento computacional. O Bebras disponibiliza tarefas voltadas para faixas etárias específicas, e o aluno tem acesso a dúvidas e soluções para exercitar essa habilidade e entender melhor os mecanismos envolvidos no processamento da informação e na resolução de problemas.

2.2.2 *Pensamento Abstrato*

Em primeiro lugar, é essencial analisar os aspectos do pensamento abstrato e como eles impactam os processos de absorção pelo aluno. Além disso, analisar as vantagens e desvantagens da generalização. As vantagens incluem a facilidade de resolver problemas complexos, a construção real de linhas de raciocínio e pensamento divergente (pensar de outras maneiras ou direções para resolver o mesmo problema) (PARK; HYUN, 2014). Em outros casos, a generalização excessiva e quando conduzida para os piores cenários pode agravar os sintomas de depressão ou ansiedade, além de interferências e limitações nos casos de transtorno ou transtorno mental (PARK; HYUN; HEUILAN, 2015).

O pensamento abstrato é a capacidade de isolar um problema e resolvê-lo separadamente e funciona construindo uma sequência de etapas para atingir um objetivo específico (STATTER; ARMONI, 2020). O elemento crítico dessa habilidade é aplicar a técnica de ignorar os detalhes inicialmente e pensar macro sobre o problema para chegar a uma solução mais transparente e eficiente. A capacidade de abstrair um problema também é uma das principais aliadas no ensino de programação, pois motiva o aluno a construir sua linha de raciocínio para chegar a uma solução e permite traduzir pseudocódigo para qualquer linguagem, entendendo sua sintaxe e semântica com maior facilidade.

Entre as ferramentas que estimulam o pensamento abstrato está o CODE, que possui um catálogo de cursos em que algumas tarefas estão disponíveis para crianças e adolescentes (ou alunos iniciantes) exercitarem alguns tópicos introdutórios e essenciais à programação, como resolução de problemas e pensamento abstrato. Além desta ferramenta, existem diferentes formas de trabalhar essa habilidade, uma delas é desenvolver jogos interativos para esse fim, utilizando motores de jogo como Unity e Construct, por exemplo.

No entanto, uma metodologia mais subjetiva também combina a abordagem hierárquica do PGK com o *framework* de Armoni (STATTER; ARMONI, 2020). Esta metodologia serve de suporte fundamental na programação, pois ao explorar de forma mais extensa e contínua, o programador vai desvendar o problema, o que permite um melhor conhecimento e contato com o assunto.

2.2.3 *Raciocínio Matemático*

As duas habilidades anteriores estão mais associadas aos procedimentos usados para buscar uma solução para um determinado problema de programação e da vida cotidiana. A

competência de raciocínio matemático está relacionada às questões mais analíticas e lógicas dessa estrutura (ATTALLAH; ILAGURE; CHANG, 2018). Alguns conceitos matemáticos, como indução, inferência e raciocínio lógico, são essenciais para a programação de computadores. Além disso, compreender construções, notações, operadores, fórmulas, regras e formalidades matemáticas e lógicas agregam valor à sequência de instruções definida pelo programador e evita bugs e erros no código (DRACHOVA et al., 2015).

A principal metodologia utilizada para este fim é proporcionada pela metacognição do aprendiz e pela utilização de recursos de linguagens que oferecem um conjunto de funções intrínsecas, mas, além disso, permite compreender essas funções de forma mais aprofundada. Investigar as especificações da matemática em detalhes seria outro método para alcançar melhorias nesse aspecto (GOMES et al., 2017) (SOUZA et al., 2019). Esta investigação pode começar do mais básico ao mais avançado e deve incluir temas fundamentais da matemática, como operações, operadores, métodos, funções, recursão, sequências, teoremas e modelos (SOUZA et al., 2019). O conhecimento e domínio destes conteúdos favorecem a otimização do processo de pensamento lógico e a elucidação de conceitos matemáticos.

2.2.4 *Flexibilidade Cognitiva*

Flexibilidade cognitiva é a capacidade de guiar pensamentos e classificar soluções divergentes na resolução de problemas, analisando soluções menos óbvias para um problema (DURAK, 2020). Assim, diferentes estratégias comportamentais podem ser encontradas neste cenário, o que permite a delimitação de várias alternativas de solução com o mesmo propósito. Essas novas alternativas podem facilitar a busca por soluções mais eficazes e transparentes, aumentando o entendimento desses problemas.

Porém, essa habilidade não é adequada para determinados grupos de pessoas, ou seja, em casos com baixa adaptabilidade. Esses casos referem-se a pessoas com algumas deficiências intelectuais, como o espectro autista. Pessoas com espectro autista tendem a depender de uma rotina, o que implica em baixa adaptabilidade às mudanças, principal característica da flexibilidade cognitiva. Nessa perspectiva, isso significa que qualquer mudança nos caminhos de resolução de problemas para as pessoas com esse transtorno acarreta dificuldades na identificação dessas soluções e ambiguidade.

Para fomentar esta competência, no âmbito da programação, os professores devem introduzir novos métodos de ensino para que os alunos possam melhorar o seu desempenho. Outro ponto pertinente é que o desenvolvimento da flexibilidade cognitiva depende da prática e absorção de conhecimentos sobre as etapas que a compõem (DURAK, 2020).

Portanto, estar aberto a outros pontos de vista, ser tolerante com os erros (o que resulta em novas formas de pensar e examinar problemas), encorajar a dinâmica de grupo para encorajar "pensar fora da caixa"(como o *brainstorming*¹), fomentar opiniões diferentes e estimular criativi-

¹ **Brainstorming** é uma técnica para estimular o surgimento de soluções criativas. A Tempestade de Ideias, em português, é feita em uma reunião e permite o compartilhamento de ideias, soluções e insights valiosos para a

dade são algumas dessas etapas que contribuem positivamente para uma flexibilidade cognitiva sólida e maior fluidez de raciocínio.

2.3 Usabilidade

Usabilidade é um termo usado para definir a facilidade com que as pessoas podem empregar uma ferramenta ou objeto para realizar uma tarefa específica e essencial. Pode-se referir também à medição dos princípios por trás da eficiência percebida de um objeto. Facilidade de aprendizado, uso eficiente e satisfação são pontos fundamentais para que o usuário perceba uma boa usabilidade. A usabilidade pode ser medida por meio do produto ou ambiente testado por um grupo de pessoas que o utiliza para atingir o objetivo específico, principalmente com seu nível de eficácia, eficiência e satisfação de um determinado grupo de usuários (ROCHA; PADOVANI, 2016).

Nesse caso, mostra que a usabilidade durante o desenvolvimento do produto pode ser fundamental para o resultado final, enquanto métodos ágeis que integram estratégias são utilizados durante sua construção. A usabilidade pode ser significativa, pois é responsável por executar com sucesso tarefas específicas.

Diante dos fatos apresentados acima, é essencial analisar a usabilidade de uma aplicação informatizada, pois garante uma mensuração da agradabilidade entre os usuários durante o processo de aplicação de um instrumento. Entre as técnicas existentes, existe o modelo de USE (DAVIS, 1989) (LUND, 2001) que aponta quatro critérios em relação às dimensões, a saber:

- **Utilidade:** verificar a opinião dos participantes quanto à utilidade e eficácia do instrumento em relação aos seus objetivos;
- **Facilidade de uso:** obter a percepção do participante sobre a facilidade de uso e a disponibilidade do instrumento;
- **Facilidade de aprendizado:** conhecer a opinião dos participantes sobre a facilidade de aprendizado do instrumento;
- **Satisfação:** medir o nível de satisfação dos participantes usando as instruções contidas no instrumento.

3 METODOLOGIA

Neste capítulo, apresentamos o planejamento dos estudos conduzidos para contemplar o objetivo desta pesquisa. Ao final desta pesquisa, queremos saber se o instrumento feito, a partir do teste em papel-e-lápis, provê uma usabilidade melhor no rastreamento das habilidades preditoras de programação. Para responder a esta pergunta e concluir este estudo, foram necessárias três etapas:

- **Etapa 1 - *Screening Programming*.** Sistematizamos o instrumento em papel-e-lápis com base em uma abordagem metodológica para concepção e desenvolvimento do trabalho (RIBEIRO, 2012);
- **Etapa 2 - *Elaboração do Questionário*.** Elaboramos um questionário exploratório supervisionado como base o modelo USE (DAVIS, 1989) (LUND, 2001), que foca as seguintes propriedades: utilidade, satisfação, facilidade de usar e facilidade aprender;
- **Etapa 3 - *Aplicação do Questionário*.** Aplicamos um questionário exploratório supervisionado junto aos estudantes de um CS1, da cidade de Campina Grande-PB, a fim de coletar informações sobre a usabilidade de cada um dos cenários de estudo. Logo em seguida, analisamos os resultados obtidos da aplicação, visando verificar a usabilidade do *Screening Programming*.

3.1 *Screening Programming*

Nesta seção, apresentamos o planejamento da concepção e desenvolvimento do sistema *Screening Programming*. E respondemos à seguinte pergunta de pesquisa:

- **QP1.** Qual metodologia de desenvolvimento é utilizada para instrumentalizar um teste em papel-e-lápis?

As aplicações multimídias são, em sua essência, sistemas. Desta forma, um projeto multimídia é equivalente a um produto de software (levando em consideração ao seu desenvolvimento) (RIBEIRO, 2012). Assim, as fases de desenvolvimento de um projeto multimídia obedecem a uma metodologia que divide as tarefas a realizar em quatro fases principais, a saber: análise de planejamento, design, desenvolvimento e, validação e arranque.

3.1.1 *Análise de Planejamento*

Nesta fase, existem três tarefas principais:

- **Lançamento do projeto:** a definição do âmbito do trabalho é a primeira atividade de gestão do projeto (PRESSMAN; MAXIM, 2016). Isto garante que o produto desenvolvido é o produto solicitado pelo cliente. Portanto, esta tarefa inicial teve os seguintes objetivos principais:

- Confirmar o âmbito do trabalho no contexto do projeto “Mensuração de habilidades cognitivas introdutórias de programação por meio de uma avaliação adaptativa informatizada”;
 - Definir as metodologias utilizadas na realização dos trabalhos propostos;
 - Identificar a equipe de trabalho;
 - Realizar a reunião de *kick-off*¹ a fim de formalizar o início dos trabalhos. Devido à natureza multidisciplinar deste tipo de desenvolvimento, requereu o envolvimento de pesquisadores em psicometria, estatística e engenharia informática; e ao fato da pandemia mundial do coronavírus, foi necessária a realização de reuniões de *kick-off*.
- **Produzir o plano de projeto:** nesta tarefa, elaboramos e avaliamos o plano de trabalho detalhado; e atribuímos as responsabilidades a cada membro da equipe da pesquisa. De acordo com o plano de projetos, foram necessários dois semestres para a conclusão desta pesquisa.
 - **Desenvolver os requisitos do sistema:** em conjunto com a equipe de projeto, desenvolvemos um entendimento básico dos princípios chaves do sistema, bem como o fluxo de informações sobre os requisitos e funcionamento do sistema. Esta tarefa foi fundamental para alinhar as expectativas e a mobilização da equipe para conclusão do projeto. Neste contexto, surgiu a necessidade de desenvolver um sistema web além da informatização dos instrumentos de avaliação inicialmente proposto para este estudo. Este sistema permitirá a integração de novos instrumentos em trabalhos futuros.

3.1.2 Design

Nesta fase, existem quatro tarefas principais:

- **Desenhar a arquitetura do sistema:** Com base nas informações que levantamos nas tarefas anteriores, conseguimos desenhar a arquitetura do sistema e definir seus requisitos funcionais e técnicos. Também foi possível o desenvolvimento do modelo lógico da base de dados com a identificação de todas as entidades necessárias para manter as respectivas relações funcionando no sistema.
- **Desenhar as interfaces do sistema:** o objetivo desta tarefa é conceber uma interface simples que exige o menor esforço de aprendizagem por parte do utilizador. Com base em uma boa interface, os utilizadores terão um modelo mental claro das interfaces que lhe permitirá prever com segurança o que vai acontecer em resposta às suas ações (SHNEIDERMAN et al., 2016). Por tanto, fizemos o levantamento dos seguintes elementos chave da interface humana:

¹ O *kick off* é uma reunião realizada na fase inicial de um projeto com o intuito de alinhar todos os detalhes entre os *stakeholders* envolvidos — o cliente, o gestor do projeto e os membros da equipe, por exemplo.

- Características dos Usuários;
 - Discriminação dos principais tipos de interface (público e após autenticação);
 - Modelo de navegação;
 - Arquitetura das páginas;
 - Princípios comuns de design e de formatação de informação nas páginas.
- **Elaborar os itens dos instrumentos:** esta tarefa teve como objetivo a construção dos itens que constituem a base de dados que será utilizada para construir os instrumentos. Cada item é composto pela descrição e suas alternativas.
 - **Desenvolver plano e estratégia de validação da aplicação:** esta tarefa teve como objetivo sistematizar os fluxos e as unidades funcionais do sistema, de forma a permitir a identificação dos casos de teste necessários para avaliar:
 - A qualidade tecnológica relacionada à disponibilidade do sistema, tempo de respostas às tarefas e a sua segurança;
 - O funcionamento correto dos componentes do sistema;
 - A adequação das funcionalidades a fim de saber se foram atendidas;
 - A coerência do sistema entre as funcionalidades. Com o suporte a este trabalho, verificamos se a aplicação final correspondeu aos objetivos elencados, se funcionou de forma adequada nas plataformas escolhidas, e se foi de encontro às necessidades do utilizador final.

3.1.3 *Desenvolvimento*

O desenvolvimento do sistema levou em consideração os requisitos que foram detalhados nas etapas anteriores. Para tanto, dividimos estas etapas em protótipos, garantindo uma melhor interação ao longo da fase de desenvolvimento, a colaboração dos membros da equipe, a construção do sistema com uma baixa taxa de erro, e conseqüentemente redução de tempo na fase de Validação e Arranque. Desta forma, tivemos a necessidade de garantir as seguintes tarefas:

- Capturar o feedback (erros, inconformidades, sugestões) dos protótipos que desenvolvemos nesta etapa;
- Planejar as funcionalidades e incluí-las em cada protótipo levando em consideração os limites de tempo de cada fase de desenvolvimento;
- Desenvolver as especificações detalhadas, para a inclusão de novas funcionalidades previstas para o protótipo;

- Adaptar as especificações para a correção e melhorar as funcionalidades implementadas em protótipos anteriores;
- Validar, com testes unitários, integrados, e de interface pessoa máquina do protótipo.

3.1.4 *Validação e Arranque*

Nesta fase, existem duas tarefas principais:

- **Validação do sistema:** após a conclusão do desenvolvimento do sistema, iniciamos a etapa de testes que tiveram como objetivos realizar validação de forma integrada da qualidade técnica do sistema e da sua adequação. Para tanto, realizamos as seguintes tarefas:
 - Organizar uma base de dados sobre os conteúdos dos testes;
 - Realizar os testes de integração com objetivo de validar a qualidade técnica do sistema desenvolvido e os tempos de resposta;
 - Validar o funcionamento do sistema e alinhar com os objetivos definidos no início do projeto;
 - Corrigir os erros técnicos encontrados;
 - Corrigir os erros funcionais e de interface pessoa-máquina encontrados.
- **Formação e Mobilização:** esta tarefa tem como objetivo preparar os professores e alunos a utilizarem o sistema. Esta tarefa foi fundamental para o sucesso da etapa seguinte. Esta formação decorreu durante uma semana, dividida entre sessões à distância, seguida de utilização experimental da plataforma no dispositivo móvel com suporte às dúvidas.

3.2 **Elaboração do Questionário**

Para cumprir o objetivo geral deste trabalho, concebemos/desenvolvemos um questionário aos participantes envolvidos na aplicação do sistema web *Screening Programming* para obter informações sobre a sua usabilidade. Nesta seção, apresentaremos como respondemos à seguinte pergunta de pesquisa:

- **QP2.** Quais métricas são utilizadas para avaliar usabilidade?

Adotamos às métricas do modelo USE (DAVIS, 1989) (LUND, 2001) que avalia a usabilidade em quatro critérios, a saber: utilidade, facilidade de uso, facilidade de aprendizado e satisfação. Elaboramos um questionário para obtenção dessas informações contém 16 questões, conforme o quadro 4. Disponibilizamos em formato eletrônico e coletamos os dados necessários para complementar o objetivo desta pesquisa.

Quadro 4 – Indicadores de dimensão do modelo USE

Métrica	Questão
Utilidade	O <i>Screening Programming</i> é útil?
	O <i>Screening Programming</i> é eficaz?
	O <i>Screening Programming</i> melhora a avaliação das habilidades preditoras em tarefas de programação?
	O <i>Screening Programming</i> economiza tempo?
Facilidade de uso	O <i>Screening Programming</i> é fácil de usar?
	O <i>Screening Programming</i> pode ser usado sem instruções escritas?
	O <i>Screening Programming</i> pode ser usado com sucesso todas as vezes?
Facilidade de aprendizado	Aprendi a usá-lo rapidamente?
	É fácil lembrar como usar o <i>Screening Programming</i> ?
	Rapidamente me tornei especialista em usar o <i>Screening Programming</i> ?
Satisfação	Estou satisfeito com o <i>Screening Programming</i> ?
	Eu recomendaria <i>Screening Programming</i> a um amigo?
	O <i>Screening Programming</i> funciona o que eu quero?
	O <i>Screening Programming</i> é agradável de usar?
	O <i>Screening Programming</i> é divertido?
	O <i>Screening Programming</i> é adequado para seus usuários?

Fonte: Elaborada pelo autor (2022).

Adotamos um modelo de escala Likert de cinco pontos de formulários de avaliação na Pesquisa, com a resposta a cada questão representada da seguinte forma: 1 = discordo, 2 = discordo parcialmente, 3 = indeciso, 4 = concordo parcialmente e 5 = concordo totalmente. Com base nas notas atribuídas, faremos uma média das respostas para cada questão.

3.3 Aplicação do Questionário

Por fim, aplicamos um questionário aos participantes envolvidos na aplicação do sistema web *Screening Programming* para obter informações sobre a sua usabilidade. Nesta seção, apresentaremos como respondemos à seguinte pergunta de pesquisa:

- **QP3.** O sistema desenvolvido apresenta uma boa usabilidade?

Para a realização deste estudo, foi necessário aplicar os questionários a todos os participantes que tiveram a experiência de uso do *Screening Programming* para estimar as habilidades

cognitivas preditoras de programação. Capturamos a satisfação do participante com as diferentes dimensões e respectivos indicadores em análise.

3.3.1 Preparação

Não foi necessário adquirir nenhuma ferramenta para a realização deste levantamento, pois tudo o que foi proposto foi desenvolvido pelos próprios pesquisadores. Porém, os seguintes equipamentos foram necessários para esta Pesquisa: i) dispositivo móvel ou computador com acesso à Internet; e, ii) formulários de avaliação online.

3.3.2 Análise dos Dados

Os dados coletados nesta análise são numéricos, decimais e classificados quantitativamente em uma escala de razão. Apresentamos esses dados por meio de uma análise estatística descritiva na sua interpretação e aplicado o teste de proporção para verificar se 50% ou mais dos usuários “concordam parcialmente” ou “concordam totalmente” que o sistema obedece a característica questionada. Teste de proporção é uma estimativa da análise de dados que envolve somente duas categorias, como aprovação ou reprovação. Comparamos a proporção dos usuários com a proporção hipotética especificada no questionário de avaliação.

Realizamos o teste de proporção por meio da ferramenta *Action Stat* que utiliza a proporção de uma única amostra para gerar dados com indicação de sucesso. Para aplicar o teste de proporção de amostra única é necessário transformar os dados que estavam na escala de *likert* em dicotômicos (ou seja, atribuir 0 para os indicadores abaixo de 4 e atribuir 1 para aqueles profissionais que indicaram 4 ou 5, ou seja, concorda parcialmente ou concorda totalmente que a ferramenta atende o requisito de usabilidade). Depois que a entrada dos dados é fornecida na aplicação, o tipo de teste a ser utilizado é TCL (teste com aproximação normal e com correção de continuidade) com o nível de significância de 95%, hipótese nula em 50% e hipótese alternativa ‘maior que’, pois, queremos aceitar que 50% ou ‘mais’ dos usuários concordam totalmente ou parcialmente que o sistema atende à métrica de usabilidade. Nos resultados obtidos, se o valor de *p* é menor que 0,05 de relevância, então pode se afirmar que a hipótese nula pode ser rejeitada.

3.3.3 Análise às ameaças

Consideramos alguns fatores que podem gerar ameaças e influenciar diretamente na conclusão deste trabalho, entre eles

- Os participantes da pesquisa podem se sentir intimidados ou desconfortáveis ao fazer os testes. Aplicamos as diretrizes do comitê de ética em pesquisa para minimizar esse possível constrangimento. O Comitê de Ética em Pesquisa com Seres Humanos da Universidade Federal de Campina Grande (UFCG) aprovou esta pesquisa (Protocolo: 55160816.6.0000.5182). Participaram do estudo apenas os participantes que assinaram o

Termo de Assentimento para menores de 18 anos (Anexo A) ou Termo de Consentimento Livre Esclarecido (Anexo B);

- Os instrumentos foram corrigidos automaticamente para mitigar possíveis erros humanos;
- Como toda pesquisa empírica, este trabalho tem ameaças de validade. O número de sujeitos participantes do estudo não permite a generalização dos resultados.

3.3.4 Execução da Pesquisa

Abaixo, descrevemos uma sequência de etapas que foram úteis para conduzir o questionário:

- Executamos apenas um questionário para cada participante. Participaram deste estudo 101 alunos oriundos de universidades locais que utilizaram o *Screening Programming* durante 4 semanas;
- Cada participante assinou um Termo de Consentimento Livre e Esclarecido para participar da pesquisa. Cada participante levou em média 10 minutos para responder aos questionários;
- Separamos e analisamos os dados e com base nos resultados obtidos, contestamos ou aceitamos a hipótese nula proposta no trabalho.

3.3.5 Divulgação

Os resultados obtidos na pesquisa e os dados coletados, planilhas e dados secundários relacionados foram cedidos em diretório aberto para visualização de outros pesquisadores da área de estudo.

4 SCREENING PROGRAMMING

Neste capítulo, apresentamos os passos dados na concepção/construção do sistema *Screening Programming*. Trata-se de um sistema que apoia os educadores de cursos superiores e técnicos da área de informática a mensurar habilidades cognitivas preditoras de programação.

4.1 Análise de Requisitos

Coletamos os requisitos do sistema junto à especialistas da área da Computação e Psicometria. Definimos um esboço de uma solução que serviu para implementar o sistema. Recolhemos essas informações por meio de uma entrevista semiestruturada com questões abertas. Consideramos a relação profissional do entrevistado com o tema nas dimensões que contemplam essa pesquisa. Os profissionais participantes assinaram o Termo de Consentimento Livre Esclarecido para participar da entrevista.

Após o término da entrevista, elaboramos uma síntese das principais informações e as organizamos em uma tabela de análise por componente. Essas informações serviram de base para a elaboração/complexidade de um projeto, por meio de um conjunto de requisitos. A seguir, detalharemos os requisitos funcionais e não funcionais do sistema.

4.1.1 Requisitos Funcionais

Os requisitos funcionais são as funcionalidades que o sistema deve fazer. Os requisitos funcionais são descritos pela equipe de forma abstrata para que o usuário possa compreendê-los. Os requisitos mais específicos descrevem em detalhes as funções do sistema, suas entradas e saídas, exceções entre outras funcionalidades (SOMMERVILLE et al., 2011). Apresentamos os requisitos funcionais do sistema que será capaz de executar:

- Disponibilizar as seguintes páginas de acesso público:
 - Página inicial do sistema com o acesso ao *login*, permitindo recolher os seguintes dados obrigatórios: e-mail e senha; se o usuário tiver cadastro com os dados do google, poderá entrar no sistema direto.
 - Página de recuperar senha, permitindo recuperar a senha por meio do e-mail.
 - Página com informações sobre o sistema;
 - Página de créditos do sistema;
 - Página de contato, permitindo recolher os seguintes dados obrigatórios: nome, e-mail, assunto e mensagem.
 - Página de cadastro do professor ou aluno, permitindo recolher os seguintes dados obrigatórios: nome, e-mail e senha; o usuário pode optar em utilizar os dados de acesso do google, de tal forma que o sistema será integrado.

- Disponibilizar as seguintes páginas de acesso pós-autenticação para o professor:
- Página de continuar cadastro do professor, permitindo recolher os seguintes dados obrigatórios: nome, foto, data de nascimento, gênero, período, cidade, uf, e-mail, senha (caso não esteja logado com a conta do google), instituição.
- Página inicial com informações dos grupos criados;
- Página de cadastro de grupo, permitindo recolher os seguintes dados obrigatórios: título do grupo e o período.
- Página para convidar um aluno ao grupo por meio de um e-mail;
- Página de acesso ao grupo criado, contendo o relatório de alunos e os instrumentos utilizados para avaliar as habilidades preditoras de programação. Além disso, acesso para editar/excluir grupo criado e excluir alunos do grupo;
- Página para acessar cada instrumento avaliativo;
- Página para acessar a estatística dos instrumentos.
- Disponibilizar as seguintes páginas de acesso pós-autenticação para o aluno:
 - Página de continuar cadastro do aluno, permitindo recolher os seguintes dados obrigatórios: nome, foto, data de nascimento, gênero, período, cidade, uf, e-mail, senha (caso não esteja logado com a conta do google), instituição.
 - Página inicial com acesso à cada instrumento avaliativo;
 - Página de resultados, exibindo o seu desempenho.
- Estabelecer os seguintes requisitos de alto nível:
 - O sistema deve estimar as seguintes habilidades:
 - * **Resolução de Problemas** - estimar a habilidade do aluno em conseguir encontrar uma solução para resolver um determinado problema;
 - * **Pensamento Abstrato** - estimar a habilidade do aluno em dividir um determinado problema em partes menores e resolvê-los separadamente;
 - * **Raciocínio Matemático** - estimar a habilidade do aluno na matemática para resolver problemas;
 - * **Flexibilidade Cognitiva** - estimar a habilidade do aluno em pensar em diferentes soluções para o mesmo problema.

4.1.2 *Requisitos Não-funcionais*

Os requisitos não funcionais podem ser definidos como serviços que não estão diretamente relacionados com os serviços específicos oferecidos pelo sistema, que pode estar relacionado às propriedades emergentes do sistema, como confiabilidade, tempo de resposta e ocupação de área (SOMMERVILLE et al., 2011). A seguir são apresentados os requisitos não funcionais do sistema que será capaz de executar.

- **Usabilidade:** este requisito fornece ao usuário uma interface simples e fácil de usar;
- **Confiabilidade:** este requisito deve garantir ao usuário uma alta disponibilidade do sistema, para não haver interrupção durante a etapa de aplicação dos instrumentos;
- **Eficiência:** este requisito deve garantir ao usuário um desempenho de alto nível;
- **Portabilidade:** este requisito faz com que o sistema seja executado nos principais navegadores da internet;
- **Segurança:** este requisito trata todas as informações em relação a senha do usuário, controle de acesso e privacidade;
- **Instruções:** este requisito proporciona todas as instruções que devem ser claras para que o instrumento seja aplicado com o aluno;
- **Aplicação:** o sistema será necessário conexão com a Internet para sua utilização.

4.2 **Modelo de Análise**

As características que serão descritas nesta subseção referem-se ao perfil do usuário, sequência de eventos do ambiente, diagrama de casos de uso e as tecnologias usadas para o seu desenvolvimento.

4.2.1 *Perfil do Usuário*

De acordo com a relevância do sistema, os atributos do usuário são:

- **Características gerais:** O sistema é destinado a professores e alunos ingressantes em CS1. O usuário deve ter noções básicas de informática para manusear o sistema. O objetivo é mensurar as habilidades preditoras de programação em alunos ingressantes em CS1;
- **Conhecimento conceitual:** O nível de conhecimento do usuário é baixo para realização das funções e o uso do sistema;
- **Estilo cognitivo:** Não se espera o aprendizado durante o processo de aplicação do sistema, uma vez que sua prática é indicada para ser realizada trimestralmente, seu objetivo é mensurar as limitações em programação de modo que os níveis de curiosidade e persistência dos usuários na realização das funções serão considerados médio.

4.2.2 Sequência de Eventos do Instrumento

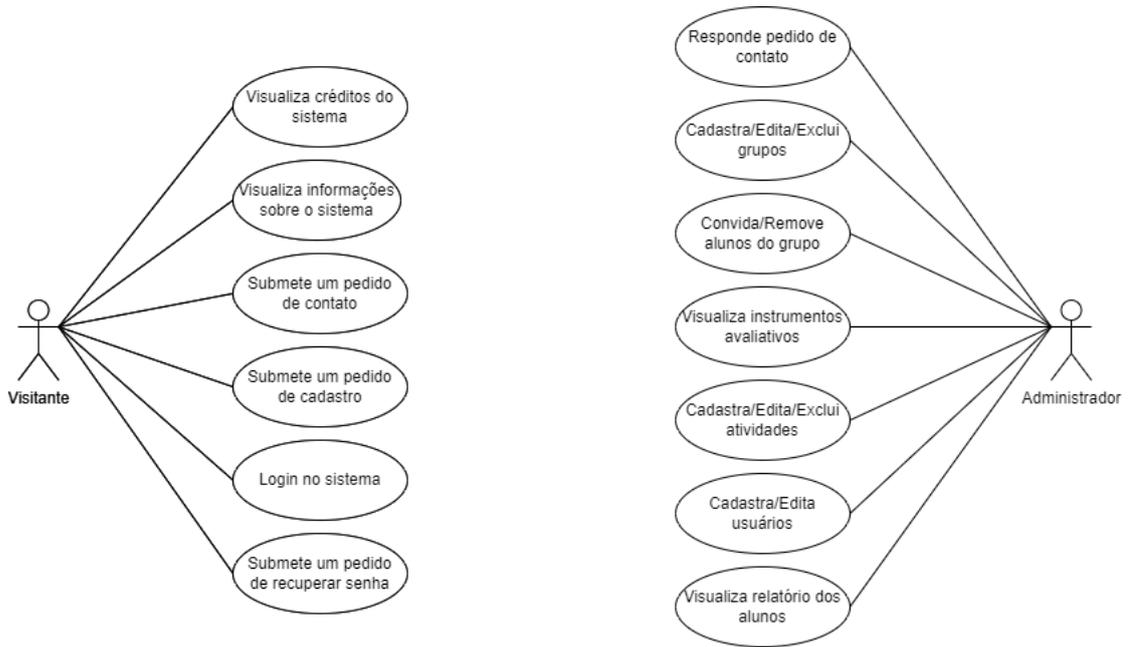
No Ensino Tradicional, professores precisam avaliar o desempenho prévio na programação dos alunos que iniciam o CS1. Muitas vezes essa etapa é ignorada pelos educadores devido ao escopo do Ensino Superior. Essa mensuração ajudará aos professores a identificar possíveis falhas ou evasão do aluno, focando em estratégias para resgatar esse aluno previamente. O *Screening Programming* surge como objetivo de estimar as habilidades predictoras de programação em alunos que iniciam um CS1, através de instrumentos de rastreamento. Ao final de todo processo será possível gerar um relatório que ajudará na avaliação do nível de habilidade do aluno, sendo esse processo realizado a qualquer momento. Estrutturamos o sistema da seguinte forma:

- **Etapa 1 - Apresentação:** o sistema inicial com a página inicial e suas funcionalidades, tais como: sobre, créditos, contatos, entrar e cadastrar uma conta. O usuário poderá efetuar o seu *login* caso tenha uma conta já cadastrada, caso contrário precisará se cadastrar. Ao se cadastrar e efetuar *login*, o usuário continuará o seu cadastro;
- **Etapa 2 - Estabelecimento das atividades:** o professor terá acesso a cadastrar/editar/excluir grupos, convidar/remover alunos dos grupos e a visualizar os instrumentos avaliativos;
- **Etapa 3 - Mensuração das habilidades:** o aluno terá acesso a responder os itens presentes em cada instrumento avaliativo e visualizar o seu desempenho;
- **Etapa 4 - Relatório final:** ao finalizar o processo de mensuração do instrumento, o professor terá a possibilidade de visualizar o relatório com os resultados do desempenho dos alunos.

4.2.3 Diagrama de Caso de Uso

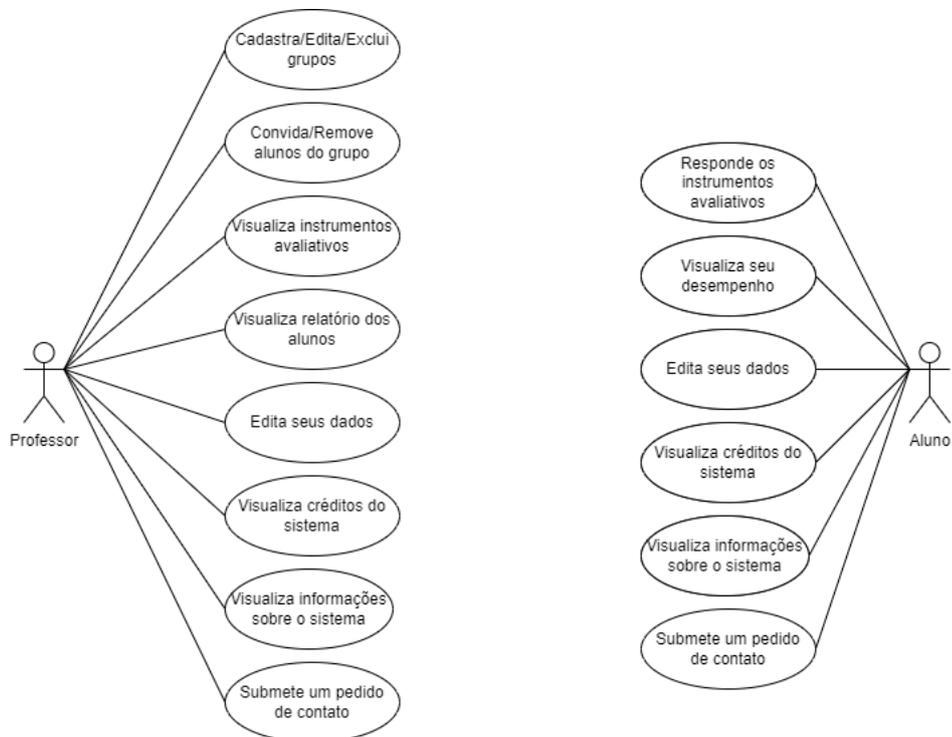
Nas Figuras 1 e 2 representamos os autores do sistema: visitante, administrador, professor e aluno. O visitante visualiza as informações sobre o sistema, créditos e submete um pedido de contato, pedido de cadastro (professor/aluno), *login* do sistema e recuperar senha. O administrador responde o pedido de contato. O professor cadastra/edita/exclui grupos, convidar/remover alunos dos grupos e a visualizar os instrumentos avaliativos e relatório com o desempenho dos alunos. O aluno responde os itens presentes em cada instrumento avaliativo e visualiza o seu desempenho.

Figura 1 – Diagrama de Caso de Uso dos Usuários Visitante e Administrador



Fonte: Elaborada pelo autor (2022).

Figura 2 – Diagrama de Caso de Uso dos Usuários Aluno e Professor



Fonte: Elaborada pelo autor (2022).

4.2.4 Tecnologias de Suporte

Nesta seção, descrevemos as tecnologias utilizadas para desenvolver o sistema *Screening Programming*.

Para o *Front-end*¹, utilizamos a linguagem JavaScript por se tratar de uma linguagem de programação interpretada que suporta vários elementos de sintaxe de programação estruturada da linguagem C. JavaScript tem semelhança com o HTML que é interpretado pelos browsers. Esta linguagem é fundamentalmente relevante para este projeto por permitir melhorar a interação do usuário por meio de uma interface intuitiva.

Para o *Back-end*² utilizamos o Nodejs³ que é um ambiente de execução para criar aplicações *Javascript* sem depender de um *browser* (navegador de internet) para a execução. A plataforma preenche uma necessidade específica neste projeto, uma vez que, almejamos a criação de aplicativos de rede altamente rápidos, escaláveis e com grande volume de conexões simultâneas.

Utilizamos *Integrated Development Environment Visual Studio Code* que é um editor de código-fonte desenvolvido pela Microsoft para Windows, Linux e macOS. Visual Studio Code⁴ oferece suporte para depuração, controle de versionamento Git incorporado, realce de sintaxe, complementação inteligente de código entre outras funcionalidades que permitiu uma melhor produtividade na equipe de desenvolvedores.

O sistema é integrado com um banco de dados MySQL (um sistema de gestão de base de dados open source da ORACLE). Utilizamos o MySQL para guardar conteúdo da aplicação, como também, os dados recolhidos durante a aplicação dos instrumentos.

4.3 Modelo de Arquitetura

Esta seção especifica o funcionamento da arquitetura do *Screening Programming*, apresentando o modelo escolhido do sistema e de seu armazenamento. A Figura 3 apresenta a arquitetura do sistema, subdividido em três partes: interface com usuário, mecânica e base de dados, foram criados os mesmo para integração entre si.

No sistema existem quatro usuários: administrador, aluno, professor e visitante. Na subseção 4.2.3 descrevemos as responsabilidades de cada usuário. A interface é formada por todas as funções que são fundamentais para a interação entre usuário e o sistema. Isto é, na interação dos usuários com o sistema se dá por meio de *touch screen* (tela sensível a toque) ou clique do mouse (periférico), fazendo com que execute as funções propostas no sistema. Concebemos a mecânica do sistema por meio de um conjunto de convicções de forma clara e determinada, para

¹ Tudo que está ao alcance e na interface que o usuário está tendo contato direto, ou seja, podendo interagir, é feito pelo *Front End*

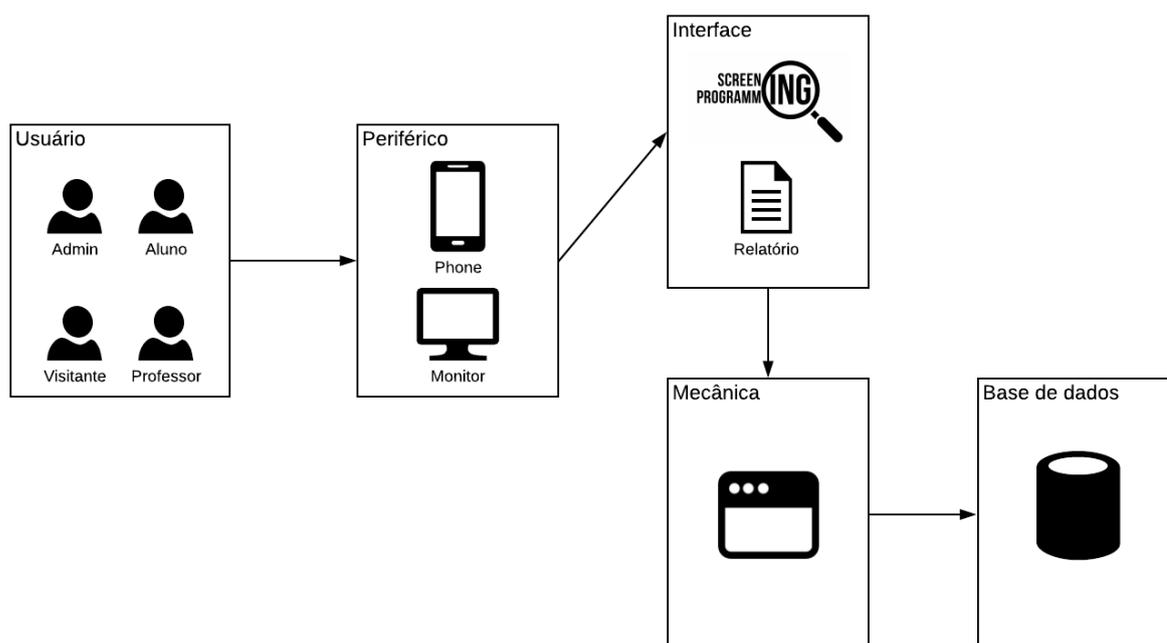
² Em muitos sites e soluções, o usuário interage diretamente com uma interface, mas os comandos que permitem a atividade acontecem por trás. Dessa forma, o *Back End* é tudo que não é visto na interface.

³ O nodejs oferece suporte para o desenvolvimento do backend. Disponível em: <https://nodejs.org/en/>

⁴ O vscode oferece suporte para desenvolvimento html e css(frontend). Disponível em: <https://code.visualstudio.com/>

que o usuário possa ter uma compreensão melhor do sistema como é representado pela sequência de eventos (subseção 4.2.2). O modelo lógico de dados foi concebido por meio do MySQL, com intuito de manter e segurar todos os requisitos que foram definidos durante todo o andamento do projeto em desenvolvimento.

Figura 3 – Arquitetura do *Screening Programming*.



Fonte: Elaborada pelo autor (2022).

4.4 Modelo de Projeto

Baseado no modelo de análise, nesta seção apresentamos a modelagem do sistema. Possibilita mostrar de forma padronizada as interfaces que apresentaremos a seguir.

4.4.1 Pré-autenticação

A Figura 4 consiste na página inicial do *Screening Programming*. Por meio desta, o usuário consegue acesso às páginas sobre créditos, contatos, cadastrar e entrar.

Figura 4 – Página inicial do *Screening Programming*.

ENTRAR

SCREENING
PROGRAMMING

Email

Senha

Entrar Cadastrar

Entrar com o google

Esqueceu sua senha?

SCREENING PROGRAMMING | 2021 - 2022
SOBRE | CREDITOS | CONTATOS

Fonte: Elaborada pelo autor (2022).

A Figura 5 é o formulário “esqueceu a senha” para acessar o sistema, composto por um campo de preenchimento obrigatório: e-mail, o botão “enviar” solicita um pedido de alteração de senha e botão para “voltar” para retornar à página inicial.

Figura 5 – Página inicial do *Screening Programming*.

ENTRAR

SCREENING PROGRAMMING

ALTERAR SENHA

INFORMAÇÕES DE RECUPERAÇÃO

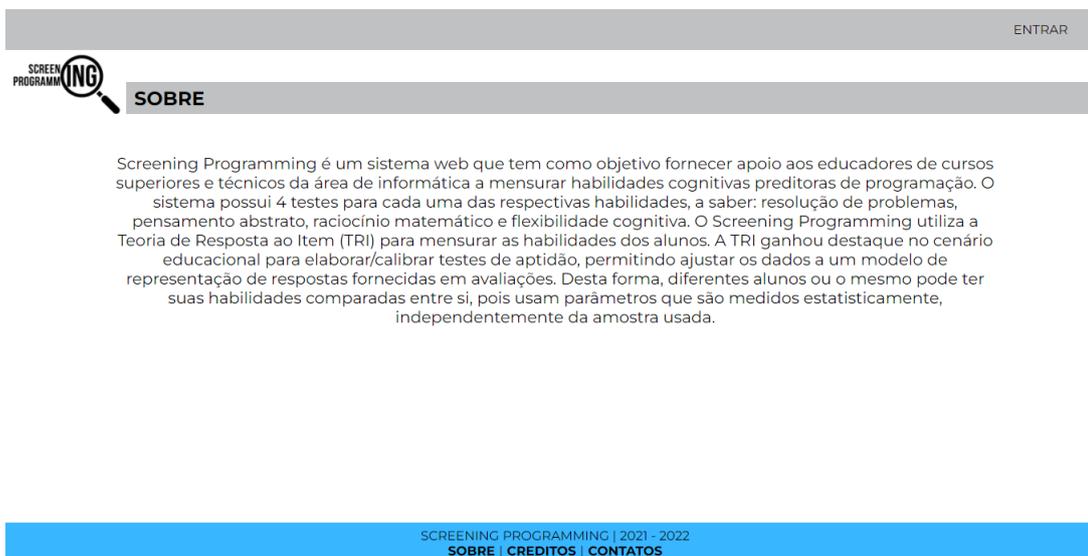
Digite o seu e-mail

VOLTAR ENVIAR

Fonte: Elaborada pelo autor (2022).

A Figura 6 consiste em informações detalhadas sobre o sistema. Após navegação até essa tela é possível voltar para a página inicial por meio do botão “entrar”.

Figura 6 – Página com informações sobre o *Screening Programming*.



Fonte: Elaborada pelo autor (2022).

A Figura 7 é constituída com informação dos respectivos parceiros para conclusão do sistema. Após a navegação até essa página é possível voltar para a página inicial por meio do botão “entrar”. Já a Figura 8 é composta por um formulário para submissão de contato com os seguintes campos de preenchimento obrigatório: nome, e-mail, assunto, caixa de texto para mensagem, o botão “enviar” para submeter o contato e o botão “entrar” para retornar à tela principal. Por fim, a Figura 9 ilustra o formulário de cadastro do usuário composto pelos seguintes campos de preenchimento obrigatório: nome, e-mail e senha. O botão “cadastrar” é feito para a confirmação do cadastro do usuário, caso contrário é só clicar no botão “entrar” para retornar à tela principal do aplicativo.

Figura 7 – Página de créditos do *Screening Programming*.



Fonte: Elaborada pelo autor (2022).

Figura 8 – Página de contatos.

ENTRAR

SCREENING PROGRAMMING

CONTATO

ENTRAREMOS EM CONTATO O MAIS RÁPIDO POSSÍVEL.
POR FAVOR PREENCHA CORRETAMENTE TODOS OS CAMPOS.

Nome*

Email*

Assunto*

Mensagem*

Enviar

SCREENING PROGRAMMING | 2021 - 2022
SOBRE | CREDITOS | CONTATOS

Fonte: Elaborada pelo autor (2022).

Figura 9 – Página de cadastro de usuário.

ENTRAR

SCREENING PROGRAMMING

Nome

E-mail

Senha

Cadastrar

Já possui uma conta?

Cadastrar com o google

SCREENING PROGRAMMING | 2021 - 2022
SOBRE | CREDITOS | CONTATOS

Fonte: Elaborada pelo autor (2022).

4.4.2 Pós-autenticação Professor

A Figura 10 ilustra o formulário de continuar cadastro do Professor. Esta página é composta pelos seguintes campos de preenchimento obrigatório: data de nascimento, gênero, período, matrícula, cidade, estado e instituição. O botão “alterar dados” é feito para confirmar a continuação do cadastro do usuário.

Figura 10 – Página de continuar cadastro do Professor.

Fonte: Elaborada pelo autor (2022).

A Figura 11 ilustra a página inicial do sistema para o Professor com informações dos grupos criados.

Figura 11 – Página inicial do Professor com informações dos grupos criados.

Fonte: Elaborada pelo autor (2022).

A Figura 12 ilustra a página de cadastro de grupos composto pelos seguintes campos de preenchimento obrigatório: título do grupo e período. A Figura 13 ilustra a página de convite dos alunos aos grupos por meio de um e-mail. O Professor fornece um e-mail e clica no botão “convidar” para enviar solicitação, caso não queira, basta clicar no botão “cancelar”.

Figura 12 – Página de cadastro de grupos.

A imagem mostra a interface de usuário para o cadastro de grupos. No topo, há uma barra de navegação com 'INICIO' e 'SAIR'. Abaixo, uma barra de menu contém 'GRUPOS' e 'MEUS DADOS'. À esquerda, há um menu lateral com 'Recrutamento' e ícones de edição e exclusão. O formulário centralizado, intitulado 'CADASTRAR GRUPO', possui dois campos de texto: 'Título do grupo' e 'Período, ex. 2021.2'. Abaixo dos campos, há dois botões azuis: 'CRIAR' e 'FECHAR'. Na base da página, há uma barra de rodapé com o texto 'SCREENING PROGRAMMING | 2021 - 2022' e links para 'SOBRE | CREDITOS | CONTATOS'.

Fonte: Elaborada pelo autor (2022).

Figura 13 – Página de convite dos alunos aos grupos.

A imagem mostra a interface de usuário para convidar alunos. No topo, há uma barra de navegação com 'INICIO' e 'SAIR'. Abaixo, uma barra de menu contém 'INICIO | ATIVIDADES | ESTATISTICAS'. À esquerda, há um menu lateral com 'Recrutamento 2021.2' e um botão 'Convidar aluno'. O formulário centralizado, intitulado 'CONVIDAR ALUNO', possui um campo de texto 'Email da aluno' e um botão 'Adicionar a lista'. Abaixo, há dois botões azuis: 'Convidar' e 'Fechar'. No fundo, há uma tabela com os seguintes cabeçalhos: 'ID', 'Matricula', 'Aluno' e 'Ações'. A tabela contém uma única linha de dados com os valores 25, 20, 24 e 29. Na base da página, há uma barra de rodapé com o texto 'SCREENING PROGRAMMING | 2021 - 2022' e links para 'SOBRE | CREDITOS | CONTATOS'.

Fonte: Elaborada pelo autor (2022).

A Figura 14 ilustra a página de acesso ao grupo, nela o professor poderá visualizar o relatório de alunos e os instrumentos utilizados para avaliar as habilidades preditoras de programação. Além disso, nesta página, o usuário pode ter acesso às funcionalidades de editar ou excluir grupos, ou excluir alunos do grupo.

Figura 14 – Página de acesso ao grupo.

ID	Matricula	Aluno	Ações
25	212080105	Arthur Pereira	
20	212080156	Breno	
24	191080527	LUANDERLANDY FELLIPE	
29	212080202	Lukas Christopher de souza	

1-4 of 21 < >

SCREENING PROGRAMMING | 2021 - 2022
SOBRE | CREDITOS | CONTATOS

Fonte: Elaborada pelo autor (2022).

A Figura 15 ilustra a página de acesso às estatísticas, nela o professor poderá visualizar o relatório geral do grupo por habilidade. Além disso, pode ter a estatística por item. Por fim, a Figura 16 ilustra um exemplo de Página do instrumento avaliativo da habilidade de Flexibilidade Cognitiva. Estes instrumentos são criados pelo administrador do sistema, e disponibiliza-os para ser utilizado com os alunos dos grupos. O professor define uma data de entrega para que os alunos enviem a atividade dentro do prazo estabelecido.

Figura 15 – Página de acesso às estatísticas.

9 - RESOLUÇÃO DE PROBLEMAS

RESOLUÇÃO DE PROBLEMAS

Quantidade de alunos: **21**
Quantidade de questões: **10**
Respostas: **210/210**

■ Acertos ■ Erros

LABIRINTO ESPECIAL

Alguns exploradores do espaço pousaram em um planeta vazio. Da sua nave podiam ver um labirinto com um objeto de prata desconhecido nele. Os exploradores lançaram seu robô no labirinto esperando que ele pudesse dar uma olhada mais de perto no objeto desconhecido. Infelizmente o robô quebrou durante a queda e agora só pode enviar e receber comandos ilegíveis sobre onde ir. O robô sugere quatro possíveis direções para seguir. Mesmo que as palavras das instruções estejam ilegíveis, ainda há apenas quatro palavras que são diferentes entre si, cada uma indicando norte, oeste, leste ou sul. Ao seguir as instruções, o robô caminhará para um quadrado adjacente (próximo quadrado) conforme as instruções. Que instruções os exploradores devem enviar ao robô para que ele chegue ao objeto de prata?

a) Hãr pòS pòS Hãr Hãr nãH

Fonte: Elaborada pelo autor (2022).

Figura 16 – Página para acessar cada instrumento avaliativo.

SCREENING PROGRAMMING | 2021

Fonte: Elaborada pelo autor (2022).

4.4.3 Pós-autenticação Aluno

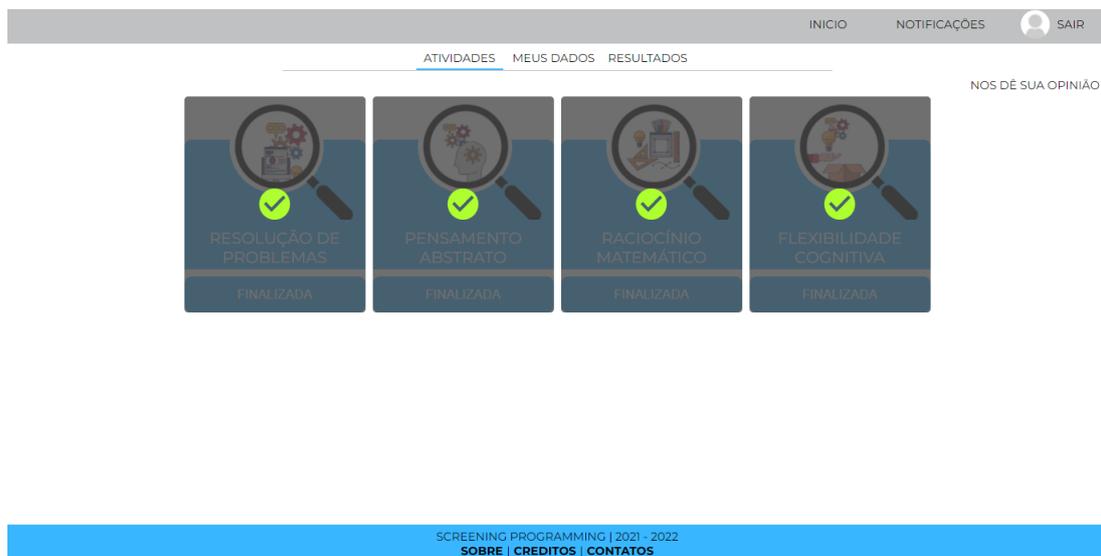
A Figura 17 ilustra o formulário de continuar cadastro do Aluno composto pelos seguintes campos de preenchimento obrigatório: data de nascimento, gênero, período, matrícula, cidade, estado e instituição. O botão “alterar dados” é feito para confirmar a continuação do cadastro do usuário.

Figura 17 – Página de continuar cadastro do Aluno.

Fonte: Elaborada pelo autor (2022).

A Figura 18 ilustra a página inicial do sistema para o Aluno com a disposição dos instrumentos avaliativos liberados pelo professor.

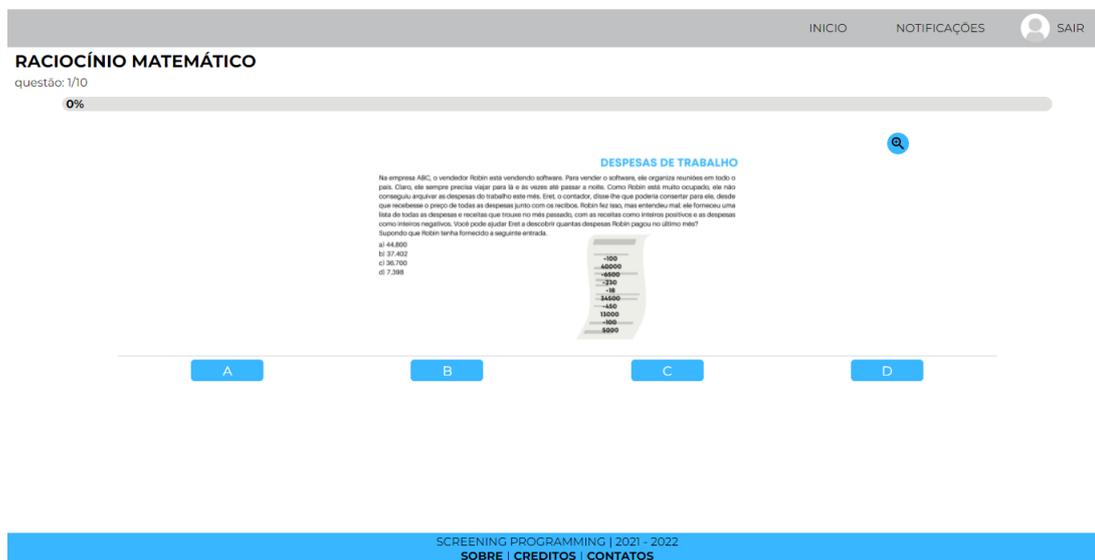
Figura 18 – Página inicial do Aluno com informações dos instrumentos disponíveis.



Fonte: Elaborada pelo autor (2022).

A Figura 19 ilustra a página inicial da aplicação de um instrumento avaliativo para o aluno. O aluno responderá cada item até que o instrumento seja concluído.

Figura 19 – Página de aplicação do instrumento avaliativo.



Fonte: Elaborada pelo autor (2022).

A Figura 20 ilustra a página do desempenho final do aluno, respectivamente o *quiz* (teste de múltipla-escolha), seus acertos e seu *score* (pontuação final) em cada habilidade avaliada.

Figura 20 – Página do desempenho final do Aluno.

INICIO NOTIFICAÇÕES SAIR		
ATIVIDADES MEUS DADOS RESULTADOS		
ATIVIDADE	ACERTO(%)	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

SCREENING PROGRAMMING | 2021 - 2022
SOBRE | CREDITOS | CONTATOS

Fonte: Elaborada pelo autor (2022).

5 ANÁLISE E RESULTADOS

Este capítulo discutirá os resultados obtidos da análise da usabilidade do sistema *Screening Programming* para medir as habilidades preditoras de programação. Para este estudo, foi realizado um teste de proporção com 95% de significância, verificando para cada métrica se 50% ou mais dos usuários “concordo parcialmente” ou “concordo totalmente” que o sistema atende às expectativas.

5.1 Análise da Utilidade

Na dimensão Utilidade, o objetivo foi medir a opinião dos usuários quanto à utilidade e eficácia do sistema *Screening Programming* sobre o rastreamento/evolução das habilidades preditoras de programação. Apresentamos os resultados na Tabela 1.

Tabela 1 – Análise dos resultados sobre a opinião dos respondentes para Utilidade

Característica	Sucesso%	Insucesso%	Média	Desvio Padrão	P-value
Utilidade	92,574%	7,426%	4,532	0,631	-
O <i>Screening Programming</i> é útil?	93,069%	6,931%	4,743	0,579	0,000
O <i>Screening Programming</i> é eficaz?	92,079%	7,921%	4,574	0,638	0,000
O <i>Screening Programming</i> melhora a avaliação das habilidades preditoras em tarefas de programação?	94,059%	5,941%	4,396	0,601	0,000
O <i>Screening Programming</i> economiza tempo?	91,089%	8,911 %	4,416	0,654	0,000

Fonte: Elaborada pelo autor (2022).

A partir da análise da Tabela 1, o *Screening Programming* é útil e eficaz no processo de mensuração das habilidades preditoras em tarefas de programação, apresentando uma diferença significativa entre a proporção de usuários satisfeitos. Segundo esses usuários, o sistema consegue melhorar a avaliação/acompanhamento de seu desempenho em tarefas prévia. Em relação ao tempo, os usuários relataram uma economia, uma vez que o processo de rastreamento das habilidades é feito com mais agilidade por meio do sistema que pode ser aplicado em qualquer local com acesso à conexão.

5.2 Análise da Facilidade de Usar

Na dimensão Facilidade de Usar, o objetivo foi mensurar a percepção dos usuários sobre a facilidade de uso e da disponibilidade do sistema *Screening Programming*. Apresentamos os resultados na Tabela 2.

Tabela 2 – Análise dos resultados sobre a opinião dos respondentes para Facilidade de Usar

Característica	Sucesso%	Insucesso%	Média	Desvio Padrão	P-value
Facilidade de Usar	95,049%	4,951%	4,719	0,549	-
O <i>Screening Programming</i> é fácil de usar?	94,059%	5,941%	4,881	0,477	0,000
O <i>Screening Programming</i> pode ser usado sem instruções escritas?	94,059%	5,941%	4,743	0,534	0,000
O <i>Screening Programming</i> pode ser usado com sucesso todas as vezes?	97,030%	2,970%	4,535	0,558	0,000

Fonte: Elaborada pelo autor (2022).

Com base na análise da Tabela 2, os usuários relataram que o *Screening Programming* é fácil de usar. O sistema possui uma interface minimalista, utilizando uma paleta de cores claras e funcionalidades diretas disponíveis facilmente, podendo ser usado sem instruções escritas. Desta forma, o acesso feito pelos usuários permitiu também uma sensação de ser usado com sucesso todas as vezes que necessário.

5.3 Análise da Facilidade de Aprender

Na dimensão Facilidade de aprender, o objetivo foi mensurar a percepção dos usuários sobre a facilidade de aprender o sistema *Screening Programming*. Apresentamos os resultados na Tabela 3.

Com base na Tabela 3, existe uma diferença significativa em relação à facilidade de aprender com o *Screening Programming*. Os usuários relataram aprender a usá-lo rapidamente devida às suas funcionalidades serem de fácil compreensão, consequentemente facilitou o processo mental dos usuários em memorizar as instruções, sendo assim, fácil de lembrar; de tal forma que, os usuários obtiveram, rapidamente, uma sensação em se tornar especialista no uso do sistema.

Tabela 3 – Análise dos resultados sobre a opinião dos respondentes para Facilidade de Aprender

Característica	Sucesso%	Insucesso%	Média	Desvio Padrão	P-value
Facilidade de Aprender	96,669%	3,331%	4,755	0,501	-
Aprendi a usá-lo rapidamente?	100%	0%	4,960	0,197	0,000
É fácil lembrar como usar o <i>Screening Programming</i> ?	100%	0%	5,000	0,000	0,000
Rapidamente me tornei especialista em usar o <i>Screening Programming</i> ?	90,009%	9,991%	4,307	0,647	0,000

Fonte: Elaborada pelo autor (2022).

5.4 Análise de Satisfação

Na dimensão Satisfação, o objetivo foi mensurar a satisfação dos usuários na utilização do *Screening Programming*. Apresentamos os resultados na Tabela 4.

Tabela 4 – Análise dos resultados sobre a opinião dos respondentes para Satisfação

Característica	Sucesso%	Insucesso%	Média	Desvio Padrão	P-value
Satisfação	88,448%	11,552%	4,308	0,686	-
Estou satisfeito com o <i>Screening Programming</i> ?	97,030%	2,970%	4,396	0,550	0,000
Eu recomendaria <i>Screening Programming</i> a um amigo?	86,139%	13,861%	4,446	0,730	0,000
O <i>Screening Programming</i> funciona o que eu quero?	92,079%	7,921%	4,307	0,615	0,000
O <i>Screening Programming</i> é agradável de usar?	100%	0%	4,465	0,502	0,000
O <i>Screening Programming</i> é divertido?	74,257%	25,743%	3,950	0,796	0,000
O <i>Screening Programming</i> é adequado para seus usuários?	81,188%	18,812%	4,287	0,766	0,000

Fonte: Elaborada pelo autor (2022).

Conforme a Tabela 4, os usuários estão satisfeitos com o *Screening Programming*, funcionando da forma como desejam. Também é do interesse desses usuários recomendar seu uso a algum amigo. O sistema é agradável de usar e adequado ao seu público, estimulando

sua participação em realizar o rastreamento/evolução das suas habilidades por conta própria, sentindo-se assim confortável ao cumprir seus objetivos.

O sistema é divertido, porém a taxa de aceitação entre os usuários neste quesito pode levar algumas informações a priori, a saber: o *Screening Programming* é um sistema de rastreio das habilidades preditoras de programação. Desta forma, os alunos podem se sentir avaliados em sua utilização, o que pode inibir o seu desempenho. Estratégias gamificadas podem ser uma alternativa para melhorar a diversão com o uso do *Screening Programming*.

6 CONSIDERAÇÕES FINAIS E SUGESTÕES PARA TRABALHOS FUTUROS

Este trabalho partiu da necessidade de identificar sinais de falhas em ingressantes de cursos Técnicos e Superiores nas disciplinas Introdutórias de Programação (CS1). Concebemos/desenvolvemos o Screening Programming com base em um banco de itens calibrados. Este sistema tem como objetivo rastrear previamente as habilidades de resolução de problemas, pensamento abstrato, raciocínio matemático e flexibilidade cognitiva de ingressantes em CS1.

A principal questão de pesquisa deste estudo é: o instrumento feito, a partir do teste em papel-e-lápis, apresenta uma boa usabilidade? A fim de entender melhor se a aplicação possui uma boa usabilidade, esta pesquisa obteve os seguintes resultados em cada questão específica dentro do escopo mais amplo:

- **(QP1).** Sistematizamos o Screening Programming com base em uma abordagem metodológica para concepção e desenvolvimento de trabalho (RIBEIRO, 2012);
- **(QP2).** Elaboramos um Survey exploratório supervisionado como base o modelo USE (DAVIS, 1989) (LUND, 2001), que foca as seguintes propriedades: utilidade, satisfação, facilidade de usar e facilidade aprender;
- **(QP3).** Aplicamos um Survey exploratório supervisionado junto aos estudantes de um CS1, da cidade de Campina Grande-PB, a fim de coletar informações sobre a usabilidade do Screening Programming.

Os resultados apontaram que o Screening Programming apresenta uma diferença significativa entre os usuários que se agradaram durante a sua aplicação, como apontam os dados para as métricas: “utilidade”, “facilidade de usar”, “facilidade de aprender” e “satisfação”. O que dá fortes indicativos que os usuários obtiveram uma empatia pelo rastreamento/evolução das habilidades predictoras de programação feito por meio de uma versão informatizada. Ressaltamos ainda, que os resultados alcançados nesta pesquisa se aplicam apenas para a realidade dos usuários da área em estudo e não podem ser generalizados.

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

- criar uma versão adaptativa do Screening Programming, desenvolvendo algoritmos adaptativos informatizados baseados na Teoria de Resposta ao Item para administrar os itens de forma adaptativa;
- realizar um experimento a fim de verificar se, a partir da escala construída e do banco de itens, o sistema adaptativo provê o levantamento de características que possam identificar sujeitos com fraca/forte habilidade em tarefas predictoras de programação.

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 na página 14.
- 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 2 vezes nas páginas 15 e 22.
- BASNET, R. B. et al. Exploring computer science students' continuance intentions to use kattis. *Education and Information Technologies*, Springer, v. 23, n. 3, 2018. Citado na página 14.
- CABO, C.; LANSIQUOT, R. D. Synergies between writing stories and writing programs in problem-solving courses. In: IEEE, MADRID, SPAIN. *Proceedings of the Frontiers in Education Conference (FIE)*. [S.l.], 2014. Citado na página 14.
- CHAO, P. Y. Exploring students' computational practice, design and performance of problem-solving through a visual programming environment. *Computers Education*, Elsevier, v. 95, 2016. Citado na página 14.
- 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 15 e 20.
- CHETTY, J.; BARLOW-JONES, G. Novice students and computer programming: Toward constructivist pedagogy. *Mediterranean Journal of Social Sciences*, MCSER Publishing, Rome, Italy, v. 5, n. 14, 2014. Citado na página 15.
- DAVIS, F. D. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, JSTOR, p. 319–340, 1989. Citado 4 vezes nas páginas 23, 24, 27 e 51.
- DEKHANE, S.; XU, X.; TSOI, M. Y. Mobile app development to increase student engagement and problem solving skills. *Journal of Information Systems Education*, Information Systems and Computing Academic Professionals, Wilmington, NC, United States, v. 24, n. 4, 2019. Citado na página 15.
- 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 2 vezes nas páginas 15 e 22.
- 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 na página 15.
- 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 15, 19 e 22.

FIGUEIREDO, J.; GARCIA-PENALVO, F. J. Improving computational thinking using follow and give instructions. In: ACM CÁDIZ, SPAIN. *Proceedings of the Technological Ecosystems for Enhancing Multiculturality (TEEM)*. [S.l.], 2017. Citado na página 15.

FIGUEIREDO, J.; GARCÍA-PEÑALVO, F. J. Teaching and learning strategies of programming for university courses. In: ACM LEÓN, SPAIN. *Proceedings of the Technological Ecosystems for Enhancing Multiculturality (TEEM)*. [S.l.], 2019. Citado na página 15.

FIGUEIREDO, J.; NATÁLIA, G.; GARCÍA-PEÑALVO, F. J. Ne-course for learning programming. In: ACM SALAMANCA, SPAIN. *Proceedings of the Technological Ecosystems for Enhancing Multiculturality (TEEM)*. [S.l.], 2016. Citado na página 15.

GOMES, A. et al. Student's characteristics and programming learning—a macanese perspective. In: IEEE, HONG KONG, CHINA. *Proceedings of the International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. [S.l.], 2017. Citado 2 vezes nas páginas 15 e 22.

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 15.

HOOSHYAR, D.; AHMAD, R. B.; NASIR, M. H. N. M. A framework for automatic text-to-flowchart conversion: A novel teaching aid for novice programmers. In: IEEE, BANDUNG, INDONESIA. *Proceedings of the International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*. [S.l.], 2014. Citado na página 15.

HOOSHYAR, D. et al. Flowchart-based approach to aid novice programmers: A novel framework. In: IEEE, KUALA LUMPUR, MALAYSIA. *Proceedings of the International Conference on Computer and Information Sciences (ICCOINS)*. [S.l.], 2014. Citado na página 15.

HOOSHYAR, D. et al. Flowchart-based programming environments for improving comprehension and problem-solving skill of novice programmers: A survey. *International Journal of Advanced Intelligence Paradigms*, v. 7, n. 1, 2015. Citado na página 15.

HOOSHYAR, D. et al. A flowchart-based multi-agent system for assisting novice programmers with problem solving activities. *Malaysian Journal Of Computer Science*, v. 28, n. 2, 2015. Citado na página 15.

HUEI, Y. C. Benefits and introduction to python programming for freshmen students using inexpensive robots. In: IEEE, WELLINGTON, NEW ZEALAND. *Proceedings of the International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*. [S.l.], 2014. Citado na página 15.

JAKOŠ, F.; VERBER, D. Learning basic programming skills with educational games: A case of primary schools in slovenia. *Journal of Educational Computing Research*, v. 55, n. 5, 2017. Citado na página 15.

JONES, G. B.; WESTHUIZEN, D. V. Pre-entry attributes are thought to influence the performance of students in computer programming. In: SPRINGER, CHAM. *Proceedings of the Southern African Computer Lecturers' Association*. [S.l.], 2017. Citado na página 15.

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 15.

LANSIQUOT, R. D.; CABO, C. Strategies to integrate writing in problem-solving courses: Promoting learning transfer in an interdisciplinary context. In: AMERICAN SOCIETY FOR ENGINEERING EDUCATION, SEATTLE, WA, UNITED STATES. *Proceedings of the American Society for Engineering Education (ASEE)*. [S.l.], 2015. Citado na página 15.

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 4 vezes nas páginas 15, 18, 19 e 20.

LUND, A. M. Measuring usability with the use questionnaire. *Usability interface*, v. 8, n. 2, p. 3–6, 2001. Citado 4 vezes nas páginas 23, 24, 27 e 51.

LÓPEZ, P. E. M. et al. The gobstones method for teaching computer programming. In: IEEE. *Proceedings of the Latin American Computer Conference (CLEI)*. [S.l.], 2017. Citado na página 15.

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 15.

MALIK, S. I.; COLDWELL-NEILSON, J. Impact of a new teaching and learning approach in an introductory programming course. *Journal of Educational Computing Research*, v. 55, n. 6, 2017. Citado na página 15.

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 15.

MEDEIROS, L. B. Programa de nivelamento em matemática básica: mensuração das habilidades predictoras de matemática em alunos por meio de um instrumento avaliativo. In: *Trabalho de Conclusão de Curso, Monografia (Licenciatura em Matemática) – Centro de Ciências Exatas e Sociais Aplicadas, Universidade Estadual da Paraíba*. [S.l.: s.n.], 2019. Citado na página 14.

MLADENOVÍĆ, M.; KR PAN, D.; MLADENOVÍĆ, S. Learning programming from scratch. In: *Proceedings of the International Conference on New Horizons in Education*. [S.l.: s.n.], 2017. Citado na página 15.

MORAIS, L. A. M. *Mastery Learning: uma abordagem personalizada de ensino no contexto de programação introdutória*. Tese (Doutorado) — Dissertação (Mestrado em Ciência da Computação) – Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande., 2015. Citado na página 14.

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 2 vezes nas páginas 15 e 20.

- NIEMELÄ, P. et al. Computational thinking as an emergent learning trajectory of mathematics. In: ACM KOLI, FINLAND. *Proceedings of the Koli Calling International Conference on Computing Education Research*. [S.l.], 2017. Citado 2 vezes nas páginas 15 e 20.
- PALMER, J. D.; FLIEGER, J.; HILLENBRAND, E. Javagrinder: A web-based platform for teaching early computing skills. In: AMERICAN SOCIETY FOR ENGINEERING EDUCATION, SEATTLE, WA, UNITED STATES. *Proceedings of the American Society for Engineering Education (ASEE)*. [S.l.], 2011. Citado na página 15.
- PAPADOPOULOS, Y.; TEGOS, S. Using microworlds to introduce programming to novices. In: IEEE. *Proceedings of the Panhellenic Conference on Informatics*. [S.l.], 2012. Citado na página 15.
- 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 3 vezes nas páginas 15, 19 e 21.
- 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 2 vezes nas páginas 15 e 21.
- PHILIP, M.; RENUMOL, V. G.; GOPEEKRISHNAN, R. A pragmatic approach to develop computational thinking skills in novices in computing education. In: IEEE. *Proceedings of the IEEE International Conference in MOOC, Innovation and Technology in Education (MITE)*. [S.l.], 2013. Citado na página 15.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 24.
- RHODES, M. G. Metacognition. *Teaching of Psychology*, Sage Publications Sage CA: Los Angeles, CA, v. 46, n. 2, p. 168–175, 2019. Citado na página 18.
- RIBEIRO, N. *Multimédia e tecnologias interativas*. [S.l.]: FCA-Editora Informática Lisboa, 2012. Citado 2 vezes nas páginas 24 e 51.
- ROCHA, E.; PADOVANI, S. Usabilidade e acessibilidade em smartphones. *Ergodesign & HCI*, v. 4, n. Especial, p. 58–66, 2016. Citado na página 23.
- 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 15.
- SANDS, P. Addressing cognitive load in the computer science classroom. *ACM Inroads*, ACM New York, NY, United States, v. 10, n. 1, 2019. Citado na página 15.
- SHNEIDERMAN, B. et al. *Designing the user interface: strategies for effective human-computer interaction*. [S.l.]: Pearson, 2016. Citado na página 25.
- SKALKA, J.; DRLÍK, M. Educational model for improving programming skills based on conceptual microlearning framework. In: SPRINGER, CHAM. *Proceedings of the International Conference on Interactive Collaborative Learning (ICL)*. [S.l.], 2018. Citado na página 15.

SOMMERVILLE, I. et al. Engenharia de software.[sl]. *Pearson Education*, v. 19, p. 23, 2011. Citado 2 vezes nas páginas 31 e 33.

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 15, 19, 20 e 22.

SPANGSBERG, T. H.; FINCHER, S.; DZIALLAS, S. Non-traditional novices' perceptions of learning to program: A framework of developing mental models. In: IEEE, SAN JOSE, CA, UNITED STATES. *Proceedings of the Frontiers in Education Conference (FIE)*. [S.l.], 2018. Citado 2 vezes nas páginas 15 e 21.

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 2 vezes nas páginas 15 e 21.

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 na página 15.

THEVATHAYAN, C.; HAMILTON, M. Supporting diverse novice programming cohorts through flexible and incremental visual constructivist pathways. In: ACM VILNIUS, LITHUANIA. *Proceedings of the Innovation and Technology in Computer Science Education (ITiCSE)*. [S.l.], 2015. Citado na página 15.

TOPALLI, D.; CAGILTAY, N. E. Improving programming skills in engineering education through problem-based game projects with scratch. *Computers Education*, Elsevier, v. 120, 2018. Citado na página 15.

VEERASAMY, A. K. et al. Relationship between perceived problem-solving skills and academic performance of novice learners in introductory programming courses. *Journal of Computer Assisted Learning*, John Wiley Sons Ltd, v. 35, n. 2, 2019. Citado na página 15.

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 14.

ÇOKLAR, A. N.; AKCAY, A. Evaluating programming self-efficacy in the context of inquiry skills and problem-solving skills: A perspective from teacher education. *World Journal on Educational Technology: Current Issues*, Science Park Research, Organization and Counseling, v. 10, n. 3, 2018. Citado na página 15.

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)