

Ferramentas Online na Aprendizagem de Programação de Computadores no Contexto do Ensino Remoto

Title: Online Tools for Learning Computer Programming in the Context of Remote Learning

Título: Herramientas en Línea para Aprender Programación Informática en el Contexto de la Educación Remota

Jonathan Ortiz Preuss
Instituto Federal Sul-rio-grandense – Câmpus
Venâncio Aires
ORCID: [0000-0002-0483-9896](https://orcid.org/0000-0002-0483-9896)
jonathan.preuss@redes.ufsm.br

Claudio Cleverson de Lima
Instituto Federal Sul-rio-grandense – Câmpus
Lajeado
ORCID: [0000-0003-0762-9680](https://orcid.org/0000-0003-0762-9680)
claudiolima@ifsul.edu.br

Resumo

Em cursos de Computação, Tecnologia Digitais e áreas correlatas, as disciplinas de Programação de Computadores e Algoritmos são disciplinas iniciais com um alto índice de reprovação. O ensino remoto adotado por consequência da pandemia COVID-19 trouxe novas barreiras, entre elas a necessidade de ferramentas online. Este trabalho tem como objetivo identificar, testar e analisar plataformas e ferramentas gratuitas online que podem auxiliar a contornar as adversidades enfrentadas nas disciplinas de Programação e Algoritmos no contexto do ensino remoto. A metodologia adotada foi estudo quantitativo, no qual foi inicialmente identificado um conjunto de ferramentas online que, posteriormente, tiveram seu nível de usabilidade mensurado por meio de questionários submetidos a uma turma de alunos do Curso Técnico em Informática para Internet. Os resultados obtidos identificaram que, entre as ferramentas analisadas, todas apresentam uma gama de pontos positivos e negativos, porém apenas uma delas apresentou um maior número de potencialidades que facilitam o processo de ensino-aprendizagem no contexto do ensino remoto. Ao fim da pesquisa constatou-se que há uma grande variedade de ferramentas online que podem auxiliar no ensino de Programação e Algoritmos, porém, são poucas as ferramentas com características que atendem todas as necessidades no contexto de ensino remoto.

Palavras-Chave: Ensino de Programação; Ferramenta Online; Algoritmos; Ensino Remoto Emergencial.

Abstract

In courses in Computing, Digital Technology and related areas, Computer Programming and Algorithms are initial subjects with a high failure rate. Remote teaching adopted as a result of the COVID-19 pandemic has brought new barriers, including the need for online tools. This work aims to identify, test and analyze free online platforms and tools that can help to overcome the adversities faced in Programming and Algorithms disciplines in the context of remote teaching. The methodology adopted was a quantitative study, in which a set of online tools was initially identified, which later had their usability level measured through questionnaires submitted to a group of students from the Technical Course in Computer Science for the Internet. The results identified that, among the analyzed tools, all have a range of positive and negative points, but only one of them presented a greater number of potentialities that facilitate the teaching-learning process in the context of remote teaching. At the end of the research, it was found that there is a wide variety of online tools that can help in the teaching of Programming and Algorithms, however, there are few tools with characteristics that meet all the needs in the context of remote teaching.

Keywords: Programming Teaching; Online Tool; Algorithms; Emergency Remote Teaching.

Resumen

En las carreras de Computación, Tecnología Digital y áreas afines, Programación de Computadores y Algoritmos son materias iniciales con un alto índice de reprobación. La enseñanza remota adoptada como resultado de la pandemia de COVID-19 trajo nuevas barreras, incluida la necesidad de herramientas en línea. Este trabajo tiene

Cite as: Preuss, J. O. & Lima, C. C. (2023). Ferramentas Online na Aprendizagem de Programação de Computadores no Contexto do Ensino Remoto. *Revista Brasileira de Informática na Educação*, 31, 790-813.
<https://doi.org/10.5753/rbie.2023.2867>

como objetivo identificar, probar y analizar plataformas y herramientas en línea gratuitas que puedan ayudar a superar las adversidades que enfrentan las disciplinas de Programación y Algoritmos en el contexto de la enseñanza remota. La metodología adoptada fue un estudio cuantitativo, en el que inicialmente se identificó un conjunto de herramientas en línea y posteriormente se midió su nivel de usabilidad a través de cuestionarios presentados a un grupo de estudiantes del Curso Técnico de Informática para Internet. Los resultados obtenidos identificaron que, entre las herramientas analizadas, todas presentan una gama de puntos positivos y negativos, pero solo una de ellas presentó un mayor número de potencialidades que facilitan el proceso de enseñanza-aprendizaje en el contexto de la enseñanza remota. Al final de la investigación se encontró que existe una gran variedad de herramientas en línea que pueden ayudar en la enseñanza de Programación y Algoritmos, sin embargo, existen pocas herramientas con características que satisfagan todas las necesidades en el contexto de la enseñanza remota.

Palabras clave: Enseñanza de Programación; Herramienta en línea; Algoritmos; Enseñanza Remota de Emergencia.

1 Introdução/Contexto

Os avanços acelerados da computação nos últimos anos têm mudado a relação entre a tecnologia e as pessoas e também a forma de trabalho das empresas. A forma como os sistemas operacionais, aplicativos e *sites* estão interligados com as atividades cotidianas evidencia a atual importância do setor e dos profissionais desenvolvedores de *softwares*.

Essa forte ligação entre o setor de *softwares* e a dinâmica da sociedade moderna tem gerado ao redor do globo um movimento de adição de disciplinas de tecnologia no ensino primário e secundário. Um exemplo desse movimento é percebido em países como os Estados Unidos e Reino Unido, que já incluíram na grade curricular do ensino primário, de forma obrigatória, as disciplinas de programação (Hussain et al., 2015).

As disciplinas de Programação de Computadores e Algoritmos são disciplinas iniciais em cursos de nível técnico ou superior de Computação e áreas afins, sendo esses cursos os responsáveis pela formação de programadores e engenheiros de *software*. As disciplinas de Programação e Algoritmos possuem uma gama de dificuldades como barreiras técnicas das linguagens de programação, ausência do acompanhamento regular dos discentes, dificuldade para desenvolver raciocínio lógico e outras. Essas adversidades enfrentadas por docentes e discentes tornam essas disciplinas detentoras de elevado índice de reprovação e, conseqüentemente, colaborando no processo de desistência dos alunos em início de curso (Gomes & Mendes, 2014).

Dada a necessidade do afastamento e isolamento em circunstância da saúde pública, resultante da alta periculosidade do vírus COVID-19¹, muitas atividades ao redor do mundo foram suspensas, entre elas as atividades presenciais de ensino e, para contornar essa situação, o Brasil adotou o ensino remoto. Mesmo com a adoção de novas técnicas e *design* educacional para a disciplina de programação de computadores considerando a modalidade EaD (Lima e Mattar, 2017), as dificuldades já conhecidas foram potencializadas na pandemia, acrescidas de questões como falta de equipamentos para discentes, acesso à internet e falta de formação docente para o uso de TICs (Lima, 2018; Júnior & Boguea, 2020).

Em meio a esse cenário estão os alunos da turma Curso Técnico em Informática para Internet Subsequente do Colégio Técnico Industrial de Santa Maria (CTISM) que, durante o período crítico do COVID-19, também adotaram o ensino remoto emergencial. O CTISM faz parte da Universidade Federal de Santa Maria (UFSM), instituição que está localizada em Santa Maria, a 5ª maior cidade em população do estado do Rio Grande do Sul, conforme o Instituto Brasileiro de Geografia e Estatística (IBGE, 2021).

¹ COVID-19 é o nome dado para a doença infecciosa causada pelo vírus SARS-CoV-2, o vírus causa uma síndrome respiratória aguda grave.

Recursos como metodologias ativas, plataformas² de gamificação, robótica, mediações e tutoriais e ferramentas³ *online* são utilizados para melhorar a didática e tentar transpor as dificuldades, facilitando o ensino dessas disciplinas. O uso de ferramentas e *softwares*, quando empregados para o ensino de programação, geram resultados positivos no processo de aprendizagem dos conteúdos pelos docentes (Khoury et al, 2020). Porém a maioria dos softwares disponíveis são desenvolvidos para plataforma *desktop*⁴ (Holanda et al., 2019).

Nesse contexto, esse trabalho busca responder à seguinte questão: quais plataformas online gratuitas existentes podem auxiliar os discentes a contornar as adversidades enfrentadas no ensino de algoritmos e programação no contexto do ensino remoto? O objetivo é a identificação, teste e análise de ferramentas gratuitas online que podem auxiliar os discentes e docentes a contornar as adversidades enfrentadas no processo de ensino e aprendizagem das disciplinas de algoritmos e programação. Os objetivos específicos são elencar as principais dificuldades enfrentadas em disciplinas de algoritmo e programação, elencar as ferramentas online gratuitas existentes para apoio ao ensino da programação, executar testes de usabilidade e funcionalidade das ferramentas e analisar as características e requisitos oferecidos pelas ferramentas elencadas.

O restante do trabalho está organizado da maneira que segue: A seção 2 apresenta uma relação dos conceitos teóricos e dos trabalhos correlatos utilizadas para embasar o presente artigo, A seção 3 elucida a metodologia utilizada para o desenvolvimento do presente trabalho, Já a Seção 4 estão expressos os resultados e discussões obtidas e por fim a Seção 5 apresenta as considerações finais em relação ao objeto de estudo.

2 Referencial Teórico

Ao longo desta seção são apresentados termos e conceitos teóricos fundamentais para o entendimento e desenvolvimento do presente trabalho, cujo objetivo é a identificação, teste e análise das ferramentas *online* que auxiliam o ensino de disciplinas de programação. Neste contexto, serão abordados os seguintes temas: Ensino EaD, Ensino Remoto, Estrutura das disciplinas de algoritmos e suas principais dificuldades.

2.1 Ensino EaD e Ensino Remoto

O contexto abordado neste trabalho é o ensino remoto emergencial e, para tal, é importante elucidar a diferença entre o modelo de ensino remoto e a Educação a Distância (EaD). Dada a necessidade do afastamento e isolamento em circunstância da pandemia COVID-19, muitas atividades foram suspensas, entre elas as atividades presenciais de ensino. Para contornar essa situação, o Brasil adotou o ensino remoto.

O ensino remoto no Brasil foi assegurado pela Lei nº 14.040, de 18 de agosto de 2020, que definiu uma série de normativas para que fossem executadas atividades pedagógicas em qualquer nível de ensino. Para cursos do nível superior, nível esse no qual se encontram as disciplinas objeto deste trabalho, o artigo 3º, inciso 1º, diz que é permitido utilizar tecnologias digitais para o desenvolvimento de atividades pedagógicas não presenciais com o objetivo de conclusão da

² Plataforma em computação refere-se a uma combinação de hardware e software que atua como uma base, plataformas também podem ser entendidas como outros softwares ou aplicativos que podem ser utilizados como base para o desenvolvimento de outros aplicativos, processos ou tecnologias.

³ Ferramentas são instrumentos que auxiliam a realização de uma determinada tarefa. Na computação uma ferramenta de programação é um programa ou aplicativo utilizado para apoio no processo de criação e manutenção de outros programas.

⁴ *Desktop* é um termo para designar computadores de mesa. Os *softwares* para *desktop* são desenvolvidos para computadores e *notebooks* e não precisam necessariamente de acesso à internet para funcionar.

carga horária exigida por cada curso (Brasil, 2020, p.1). Diferentemente do ensino remoto, instaurado em caráter emergencial, a Educação a Distância (EaD) é uma modalidade de ensino bem estabelecida e com uma estrutura totalmente projetada para as atividades pedagógicas sob um regime de assincronia e distanciamento físico entre professor e aluno.

Conforme apontado por Joye et al. (2020), a EaD possui características muito específicas, como uma estrutura e conteúdo planejados e desenvolvidos por professores multidisciplinares, a utilização de ambientes virtuais de aprendizagem propícios para as práticas pedagógicas e, por fim e mais importante, o perfil dos alunos, sendo um aluno com motivação e ímpeto para o desenvolvimento das atividades de ensino EaD. Entre estes estão os alunos dos cursos de Computação, Tecnologias Digitais e áreas afins.

2.2 Algoritmos e Pensamento Computacional

Pensamento computacional é um termo que descreve um processo mental cognitivo de resolução de problemas, com tais problemas podendo ser simples e estruturados ou problemas complexos e mal estruturados. O pensamento computacional é a representação da capacidade do ser humano de resolver problemas, através da capacidade de abstração, segmentação e análise do problema, de forma que a solução criada por esse processo cognitivo, pode ser executada por dispositivos de processamento de informação, resultando em um algoritmo (Shute & Asbell-Clarke, 2017). Um algoritmo computacional é uma sequência de instruções usadas pelo computador para realizar uma tarefa. Essas tarefas recebem dados para iniciar o seu trabalho, chamado de Entrada, um processo de processamento é realizado e um resultado é fornecido (Medina & Ferting, 2005). Denominada tríade de processamento, algoritmos para resolução de problemas com base nesse processo constituem a disciplina de algoritmos.

2.3 Estrutura das Disciplinas de Programação e Algoritmos

As disciplinas de Programação e Algoritmos fazem parte da grade curricular principalmente de cursos das áreas de Computação e Engenharia (Law *et al.*, 2010), porém, essas disciplinas estão presentes também em cursos como Física, Matemática, Meteorologia e outros (Bosse & Gerosa, 2017). Essas disciplinas envolvem conhecimentos sobre linguagens, paradigmas e técnicas de programação, como a disciplina de Lógica, que pode ser vista como uma ferramenta para desenvolver e organizar corretamente o raciocínio (Ferrandin & Stephani, 2005) tornando o indivíduo capaz de analisar e construir algoritmos (Rapkiewicz & Klemann, 2006).

A estrutura das disciplinas de Programação e similares é composta por três pilares: ementa curricular, metodologia e ferramentas para o processo de ensino. Em grande parte das instituições de ensino, as disciplinas de programação são executadas com metodologias convencionais, com aulas expositivas, lista de exercícios e avaliações somativas (Santos *et al.*, 2018; Junior & Moraes, 2020). Quanto às ferramentas utilizadas, existem diferentes recursos utilizados como o uso de linguagens lúdicas e visuais, tais como Scratch, Visual Programmer e Blockly, uso de pseudocódigo, como Portugol ou VisualG, uso da robótica e sistemas de gamificação (Holanda et al., 2019; Khouri *et al.*, 2020).

Em relação à composição das ementas curriculares, pode-se observar que existe um cenário em que as instituições de ensino usualmente selecionam as linguagens utilizadas nas disciplinas de programação de acordo com uma necessidade da indústria e não de acordo com elementos pedagógicos. Em uma análise sistemática das metodologias de ensino de programação (Khouri *et al.*, 2020) e as aplicações ao ensino profissional (Júnior & Boga, 2020), observa-se que existe um conjunto das linguagens mais comuns: Swift, Java, C++, Python e C. Dentre as linguagens identificadas na literatura, a linguagem C, independentemente de suas características estruturais, é a linguagem mais popular e utilizada nas disciplinas para o ensino de Programação e Algoritmos.

2.4 Dificuldades no Ensino e Aprendizagem de Programação

As dificuldades e complexidades do ensino e aprendizagem de programação são responsáveis por elevados índices de reprovação, com taxas que variam de 28% a 30% em âmbito global (Bennedsen, Caspersen, 2019; Simon *et al.*, 2019) e de 27,5% a 54,9% no contexto nacional (Bosse & Gerosa, 2017). A consequência é um movimento de desistência das disciplinas e abandono dos cursos por parte dos discentes.

As dificuldades que permeiam essas disciplinas são temas já discutidos e abordados na literatura. Existem trabalhos que investigam tais dificuldades com mais profundidade, além de abordar questões correlatas ao tema, como quais são os principais desafios e metodologias utilizadas no ensino de programação (Souza & Barbosa, 2016; Qian & James, 2018; Arimoto *et al.*, 2019; Júnior & Boga, 2020; Cheah, 2020) e quais são as dificuldades das disciplinas de programação nas perspectivas de alunos e professores (Bosse & Gerosa, 2017; Kadar *et al.*, 2021; Medeiros *et al.*, 2018). Ao analisar os trabalhos presentes na literatura, é possível organizar essas dificuldades em três grupos: dificuldades prévias à disciplina, dificuldades inerentes às linguagens de programação e dificuldades do ambiente de ensino.

Em relação às dificuldades prévias, estas são independentes da linguagem ou disciplina, e ligadas diretamente à falta de um conhecimento prévio dos alunos nos pré-requisitos para as disciplinas de Programação. Isso inclui o pouco domínio de Matemática, a falta da capacidade de abstração de problemas e falta de raciocínio lógico, o que gera dificuldade na capacidade do aluno de construir e realizar um pensamento computacional.

Já em um segundo grupo estão as dificuldades diretamente relacionadas com as linguagens de programação utilizadas nas disciplinas. Nesse conjunto estão dificuldades com a sintaxe e semântica das linguagens de programação e o domínio das ferramentas de programação. Em estudo realizado por Arimoto *et al.* (2019), os autores apresentaram uma síntese do grau de dificuldade dos alunos em relação aos conceitos e práticas mais comuns em programação (Tabela 1).

Tabela 1: Grau de Dificuldade dos estudantes em relação a conceitos e práticas mais comuns em programação.

Conceito / Prática	Nível 1	Nível 2	Nível 3	Nível 4	Nível 5
Lógica de programação	20%	29%	29%	26%	6%
Definição e Uso de Variáveis	41%	28%	22%	6%	3%
Definição e Uso de estruturas de decisão	34%	26%	22%	13%	5%
Definição e Uso de estruturas de repetição	30%	30%	23%	13%	4%
Definição e Uso de funções/métodos	23%	23%	25%	19%	10%
Passagem de parâmetros (valor/referência)	20%	16%	25%	22%	17%
Definição e uso de arrays	22%	20%	24%	18%	16%
Manipulação de strings	20%	21%	28%	19%	12%
Definição e uso de ponteiros	11%	14%	19%	22%	34%
Definição e uso de algoritmos recursivos	11%	14%	19%	22%	34%
Definição e uso de estruturas de dados	10%	17%	24%	22%	27%

Fonte: Arimoto (2019)

A Tabela 1 apresentou os conceitos observados pelos autores e suas respectivas porcentagens de dificuldade, de acordo com a experiência dos alunos analisados. A primeira coluna da tabela elenca os conceitos e práticas de programação e as colunas adjacentes são os níveis de dificuldade, que variam de de 1 a 5, sendo o nível 1 o mais baixo e nível 5 o mais alto de representação.

Por fim, o último grupo são as dificuldades que ocorrem no ambiente de ensino. O primeiro obstáculo é a formação docente. Em Lima (2018) e Berssanette & Francisco (2021), os autores observam que a maioria dos professores dessas disciplinas não têm uma formação pedagógica inicial adequada para o ensino com Tecnologias Digitais (TD). Outro obstáculo do ambiente de

ensino são as turmas heterogêneas, ou seja, existe uma disparidade no nível de conhecimento e habilidades dos alunos, pois cada indivíduo tem seu próprio ritmo de aprendizado e essa característica também é considerada uma dificuldade nas disciplinas de programação.

Como são disciplinas iniciais e com alto índice de reprovação, as turmas dessas disciplinas possuem um grande número de alunos, fato também tido como uma adversidade. Essa quantidade de alunos por turma, somada ao fator do ritmo individual de cada aluno, acaba gerando outros dois problemas: a falta de acompanhamento do aluno pelo professor, que não consegue identificar as necessidades discentes e a falta de *feedback* individual no processo de aprendizagem do aluno.

2.5 Trabalhos Relacionados

Nesta seção serão abordados os trabalhos correlatos que contribuíram para o desenvolvimento deste trabalho, com foco em ferramentas e plataformas para o auxílio no ensino das disciplinas de programação. Tais trabalhos foram encontrados através dos indexadores de pesquisa *Scholar* Google e *ACM Digital Library* e os critérios de seleção foram trabalhos que a temática trata diretamente de ferramentas e plataformas para o auxílio no ensino das disciplinas de programação *online*, e trabalhos com o maior número de citações.

2.5.1 Mapeamento das Ferramentas Utilizadas no Ensino de Programação na Educação a Distância

Em Junior & Morais (2020), os autores executam uma pesquisa bibliográfica da literatura brasileira, a partir do ano de 2015, visando identificar quais são e as características das ferramentas disponíveis para a demanda do ensino de programação na EAD. Neste trabalho foram analisadas nove ferramentas e os autores listam as características técnicas observadas conforme Tabela 2, e as funcionalidades e competências técnicas analisadas estão listadas na Tabela 3.

Tabela 2: Características Técnicas das Ferramentas.

Ferramenta	Forma de interação	Plataforma	Linguagem	Disponível em Português
Feeper	Textual	Web	Java	Sim
WH-IDE	Textual	Java	Portugol	Sim
TutorICC	Visual e Textual	Java	Pascal	Sim
Portugol Studio	Visual e Textual	Java	Portugol	Sim
AAPPW	Visual e Textual	Java	Portugol	Sim
TSTView	Visual e Textual	Web	Python	Sim
The Huxley	Textual	Web	Várias	Sim
SOAP	Visual e Textual	Web	C	Sim
Scratch	Visual e Textual	Web	Blocos	Sim

Fonte: Junior & Morais (2020).

Tabela 3: Funcionalidades e Competências Técnicas das Ferramentas.

Ferramenta	Visualização de Variáveis	Animação	Vídeo	Feedback	Adaptabilidade
Feeper	Não	Não	Não	Manual	Não
WH-IDE	Não	Não	Não	Manual	Não
TutorICC	Sim	Sim	Não	Automático	Sim
Portugol Studio	Sim	Não	Não	Manual	Não
AAPPW	Não	Não	Não	Manual	Não
TSTView	Não	Não	Não	Automático	Não
The Huxley	Não	Não	Não	Automático	Não
SOAP	Não	Não	Não	Automático	Não
Scratch	Não	Sim	Sim	Automático	Não

Fonte: Junior & Morais (2020).

De acordo com os autores, seis ferramentas se destacam: Feeper, WH-IDE, TutorICC, Portugol Studio, AAPW, TSTView, The Huxley, SOAP e Scratch. Porém, a maioria das ferramentas analisadas não utiliza as linguagens de programação mais comuns na realidade das disciplinas de programação e algumas das ferramentas de destaque não possuem sistema de acompanhamento ou *feedback* e não operam de forma *online*.

2.5.2 Avaliando Ambientes para Ensino de Programação com Suporte para o Desenvolvimento da Metacognição

O trabalho apresentado por Rodrigues *et al.* (2019), analisa seis ambientes/ferramentas gratuitos para o ensino de programação, avaliando a mobilização de estratégias de aprendizagem metacognitivas⁵. De acordo com os autores, existe uma dificuldade de encontrar ferramentas que sejam customizáveis, para atender às necessidades dos professores, e que ofereçam suporte para conteúdo de aprendizagem inteligente (SLC- *Smart Learning Content*). Ambientes SLC têm como características serem interativos e oferecem uma série de recursos de visualização, simulação, suporte à codificação e resolução de problemas com avaliação automática e *feedback*.

Os autores criaram um sistema de pontuação, sendo o Nível 0 o menor valor e Nível 2 o maior valor de pontuação para cada critério avaliativo. A Tabela 4 demonstra o resultado da comparação dos ambientes baseado no apoio da metacognição.

Tabela 4: Critérios de comparação de ambientes de programação com suporte a metacognição.

Habilidades	Critérios	TMC	Jutge.org	URI Academic	BOCA	Moodle+ VPL	Runestone
Reflexão	Diferentes conteúdos e atividades	0	0	0	0	1	2
	Customização do SLC	0	0	0	0	0	2
	Fórum, chat, grupos específicos	0	0	2	1	2	0
	Detecção de plágio	0	0	1	0	1	0
Planejamento	Agenda e lembretes de atividades	0	1	1	0	1	1
Monitoramento	Monitoramento de desempenho	0	1	0	0	0	1
	<i>Feedback</i> automático	1	1	0	0	0	1
Outros	Interoperabilidade	2	2	0	1	2	0
	Integração	1	0	1	1	2	0
	Usabilidade	1	2	2	1	2	1
	Total	5	7	7	4	13	8

Fonte: Rodrigues et al. (2019).

De acordo com a análise dos autores, a plataforma do Moodle com VPL possui a maior capacidade para o emprego de estratégias metacognitivas. As ferramentas com funcionalidades de avaliação automática e juízes online, o Jutge.org e o URI Academic são as ferramentas com maior pontuação e as plataformas TMC e BOCA são as ferramentas com menos recursos para a metacognição. Algumas ferramentas analisadas neste trabalho não operam de forma online, nem

⁵ Metacognição é um campo de estudos relacionado à consciência e ao automonitoramento do ato cognitivo, a metacognição como estratégia de ensino refere-se à capacidade de um aluno de autoanálise do seu processo de aprendizagem e adaptar seu pensamento para melhorar sua compreensão sobre o objeto de estudo.

<<https://www.educationcorner.com/metacognition/>>

prestam suporte à linguagem C. Já a plataforma Moodle com VPL demanda recurso de infraestrutura da instituição de ensino para manter a plataforma *online* e operante.

2.5.3 Mapeamento de Metodologias de Ensino-aprendizagem de Programação

Na literatura existe um grande número de trabalhos que tratam das complexidades envolvidas no ensino das disciplinas de Programação e Algoritmos, propondo abordagens, metodologias e ferramentas auxiliares de ensino. Contudo, ainda existe uma lacuna de trabalhos com foco na análise da usabilidade de ferramentas e plataformas online, considerando o contexto do ensino remoto.

Ao longo do trabalho de Khouri *et al.* (2020), os autores realizaram um mapeamento sistemático da literatura brasileira, no período entre 01 de janeiro de 2014 e 31 de março de 2020, visando identificar as metodologias de ensino de programação na educação básica, com foco no ensino médio. De acordo com os autores, a maior parte (85%) dos trabalhos analisados propõe atividades com uso do computador. Em todo o mapeamento foram identificadas uma série de 29 ferramentas digitais.

Table 5: Ferramentas digitais encontradas na pesquisa bibliográfica.

Ferramenta	Descrição
Tri-Logic	Ambiente de ensino e aprendizagem gamificado.
AGILE	
Code.org	
Problemas de Lógica	
Cod Comba	
CodeMonkey	
Light-Bot	
Monster Coding	
Construct2	Plataforma de programação visual (em blocos).
MIT App Inventor	
Blockly	
SCRATCH	
Mblock	
iVProg	
Visual Programmer(ViP)	Ambiente de programação visual através de símbolos para ensino de programação a alunos surdos.
VisualG	Ambiente para edição e execução de algoritmos em pseudo linguagem.
Edmodo	Ambiente Virtual de Aprendizagem.
Canvas	
Hour of Code	
MOODLE	Ambiente Virtual de Aprendizagem de código aberto, permite a instalação plugins.
VPL	Ambiente de programação textual online. Permite ao AVA Moodle.
URI Online Judge	Juiz - Plataforma que corrige os programas com resposta imediata.
MOSS	
ARDUINO	Ambiente de programação baseado em robótica.
Kit de Robótica LEGO	
Kit de Robótica Fischertechnik-ROBO TX Explore	
Robocode	
RoboMind	
SCRATCH para Arduino-S4A	

Fonte: Autoria própria.

A Tabela 5 apresenta as ferramentas listadas pelos autores, agrupadas de acordo com sua metodologia de operação. Os autores observaram que poucas ferramentas oferecem metodologia e ambiente para programação textual, metodologia mais comum nas disciplinas de programação.

As plataformas SCRATCH e MIT App Inventor são, respectivamente, as mais utilizadas nos trabalhos analisados, ambas empregando a metodologia de programação por blocos. A robótica é a segunda metodologia com mais propostas na literatura.

Por fim, após concluir a análise dos trabalhos abordados nessa seção foi identificado um conjunto de plataformas *online* e gratuitas que possibilitam a prática do ensino de algoritmos e programação e aquelas que se enquadram na proposta deste trabalho, foram incluídas no processo de avaliação, conforme Seção 4.

3 Metodologia

O presente trabalho caracteriza-se como um estudo quantitativo, pois tem como finalidade a análise das ferramentas *online* disponíveis que auxiliam os discentes a transpor as dificuldades das disciplinas de Programação e Algoritmos em um cenário de ensino remoto. No contexto de computação, de acordo com Wainer (2007), um trabalho de natureza quantitativa baseia-se em uma análise estatística de um conjunto de variáveis objetivas, que possibilita verificar as vantagens de um *software* em relação aos seus semelhantes, por meio de dados coletados pelos questionários.

Um questionário deve ter características de confiabilidade e validade e sua estrutura não deve gerar aversão aos participantes que o respondem. A elaboração de um questionário envolve quatro etapas: elaboração das perguntas e respostas, amostragem da população, avaliação das respostas e análise dos resultados. Para a estruturação das questões, recomenda-se que sejam questões escritas de forma simples, concisa e direta, com um posicionamento neutro e não sejam fraseadas na negativa. As respostas também devem seguir alguns critérios e serem balanceadas e, para isso, usa-se a escala de Likert (Wainer, 2007).

As questões do formulário foram criadas de acordo com a ISO 25010 e as respostas seguem a escala métrica de Likert. A escala de Likert é uma de escala de resposta psicométrica, que apresenta uma série de afirmações que possibilitam um indivíduo a emitir o seu grau de concordância em relação a um dado questionamento, já a ISO 2510 define um conjunto de quatro parâmetros para análise de usabilidade de software, os parâmetros são inteligibilidade, apreensibilidade, operacionalidade, atratividade (ISO/IEC 25010, 2020).

A lista das questões e suas respectivas opções de respostas são:

Q1 - Para avaliar a atratividade, os alunos foram questionados sobre sua percepção da interface da ferramenta, a partir da seguinte pergunta: “Considerando o layout da ferramenta, cores, fontes e a organização dos campos para escrita de código. Qual o grau de atratividade da ferramenta ?”

- 1 –Atrativo
- 2 –Largamente Atrativo;
- 3 –Totalmente Atrativo;
- 4 –Parcialmente Atrativo;
- 5 –Não atrativo.

Q2 - O grau de inteligibilidade foi analisado através da seguinte pergunta: “De acordo com suas impressões sobre as ferramentas, o quanto a ferramenta é apropriada para o auxílio no ensino de programação ?”:

- 1 –Apropriado;

- 2 –Parcialmente Apropriado;
- 3 –Largamente Apropriado;
- 4 –Totalmente Apropriado;
- 5 –Não Apropriado.

Q3 - O critério de apreensibilidade foi analisado por meio da seguinte pergunta: Foi realizada a seguinte pergunta: “Considere o seu processo de interação com a ferramenta, qual o grau de adequação das funções da ferramenta que possibilite o usuário aprender como utilizar e trabalhar com a ferramenta?”:

- 1 –Adequadas;
- 2 –Parcialmente Adequadas;
- 3 –Largamente Adequadas;
- 4 –Totalmente Adequadas;
- 5 –Não Adequadas.

Q4 - Por fim, para analisar a operacionalidade das ferramentas online, indagou-se: “De acordo com as características de adaptatividade para adequação ao estilo de aprendizagem do aluno, os mecanismos de feedback automático (indicadores de erro no código ou ao gerar o programa) emitido pela ferramenta são úteis como facilitador do aprendizado?”:

- 1 –Concordo;
- 2 –Concordo Largamente;
- 3 –Concordo Totalmente;
- 4 –Concordo Parcialmente;
- 5 –Não concordo.

Q5 - “Após a utilização da ferramenta, vocês usariam novamente e recomendariam para outros alunos/colegas?”:

- 1 –Usaria e Recomendaria;
- 2 –Usaria e Recomendaria Fortemente;
- 3 –Usaria e Recomendaria Esporadicamente;
- 4 –Não usaria e não recomendaria.

O desenvolvimento deste trabalho segue os seguintes processos metodológicos:

- a. Fundamentação teórica para abordagem dos temas: dificuldades presentes na execução das disciplinas de Programação e Algoritmos, dificuldades que permeiam o ensino remoto no contexto do Covid-19 e metodologias para o ensino utilizadas nas disciplinas de Programação e Algoritmos;
- b. Levantamento das plataformas *online* gratuitas existentes para apoio ao ensino da programação;
- c. Definição de critérios para testes e avaliações das plataformas identificadas;
- d. Execução de testes de usabilidade e funcionalidade das plataformas e análise dos resultados dos obtidos;
- e. Produção do quadro descritivo das ferramentas analisadas;
- f. Apresentação textual dos resultados obtidos.

Há grande disponibilidade de ferramentas *online* que possibilitam a prática de programação e o processo de seleção das ferramentas utilizadas nesse estudo segue cinco critérios: autonomia, gratuidade, monitoramento, elementos de *feedback* e linguagens aceitas. Tais critérios foram selecionados com base na revisão da literatura (subseções 2.4 e 2.5), reunindo um conjunto de critérios observados por autores em trabalhos correlatos.

O primeiro critério é ser uma ferramenta independente, ou seja, a ferramenta deve operar de forma autossuficiente sem necessidade de um ambiente virtual de aprendizagem (AVA). Muitas instituições de ensino não possuem um AVA com as características necessárias para o ensino de programação e, no contexto do ensino remoto, o caráter de urgência não possibilitou que as instituições adequassem seus respectivos AVA para as necessidades das disciplinas de programação.

A gratuidade é um critério extremamente importante pois, como aponta Júnior et al. (2020), recursos escassos, sejam financeiros ou materiais, é uma das barreiras que surgiram para a adoção do ensino remoto. Outro critério importante é a linguagem de programação: como visto por Khouri *et al.* (2020) e Júnior *et al.* (2020) a linguagem C está entre as mais comuns, a capacidade de operar tal linguagem é parâmetro de seleção da ferramenta *online*.

A ferramenta *online* deve possuir ao menos um método de *feedback* para os alunos, seja uma área para indicação de sucesso ou insucesso na execução dos algoritmos ou um sistema de apresentação da saída da execução de um dado algoritmo. E, por fim, o último critério de seleção é o monitoramento: a ferramenta deve possuir ao menos um método para que possibilite aos professores realizar o monitoramento e acompanhamento de seus alunos ou turmas.

O grupo participante da avaliação das ferramentas *online* testadas são alunos da turma do primeiro ano do Curso Técnico em Informática para Internet Subsequente (CTISM). A turma de alunos que participaram dos testes é mista, com alunos do gênero masculino e feminino, com idades entre 15 e 17 anos. Uma parcela dos alunos são residentes de centros urbanos (Santa Maria e cidades ao redor) e outro montante reside no interior do(s) município(s), deslocando-se diariamente para a instituição de ensino.

A amostragem dos participantes dos testes avaliativos das ferramentas é composta por um total de 30 alunos, com conhecimento prévio que varia de básico até avançado sobre informática (haja vista que já percorreram disciplinas introdutórias do assunto), porém não possuem conhecimento prévio em relação às ferramentas online testadas. Em relação ao conhecimento de programação, os alunos no momento da avaliação já haviam concluído metade da disciplina, logo, possuíam um relativo conhecimento dos conceitos envolvidos na programação de computadores.

4 Resultados e Discussão

Ao longo desta seção serão apresentados e discutidos os processos de identificação e seleção das ferramentas/plataformas para o auxílio no ensino de programação e os resultados obtidos do experimento de análise da usabilidade das ferramentas selecionadas.

Inicialmente foram selecionadas vinte ferramentas *online* com características mínimas capazes de viabilizar a prática da programação, a seleção dessas ferramentas foi baseada nos trabalhos correlatos presentes na literatura (seção 2.5) e através de buscas na ferramenta de pesquisa Google.

Após a seleção, cada uma das plataformas passou por uma análise a fim de aferir se as mesmas cumprem os cinco requisitos definidos para que a ferramenta atenda as necessidades de professores e alunos nas disciplinas de programação. A Tabela 6 apresenta a relação das ferramentas/plataformas *online* e os critérios de seleção observados. Ao final do processo de

análise, identificou-se que três ferramentas cumprem todos os requisitos, sendo essas: JDoodle, Coding Rooms e *Online GDB*.

Apesar da plataforma The Huxley atender todos os critérios de seleção, a mesma exige um processo de validação e verificação de professores e instituição de ensino; só após esse processo são liberadas funcionalidades de gestão de turmas. Dessa forma o tempo do processo de validação inviabilizou a análise da plataforma The Huxley.

Tabela 6: Ferramentas *Online* para Ensino de Programação.

Ferramenta	Critérios				
	Autonomia	Gratuidade	Monitoramento	feedback	Linguagens (C)
programiz	X	X		X	X
replit			X	X	X
oncompiler	X	X		X	X
ideone	X	X		X	X
interviewbit	X	X		X	X
codechef	X	X		X	X
jdoodle	X	X	X	X	X
techiedelight	X	X		X	X
mycompiler.io	X	X		X	X
codingground	X	X		X	X
rextester	X	X		X	X
ide.geeksforgeeks	X	X		X	X
codingrooms	X	X	X	X	X
wandbox	X	X		X	X
code.dcoder.tech	X	X		X	X
paiza.io	X	X		X	X
codiva.io	X	X		X	X
codepad.org	X	X		X	X
onlinegdb	X	X	X	X	X
thehuxley	X	X	X	X	X

Fonte: Autoria própria.

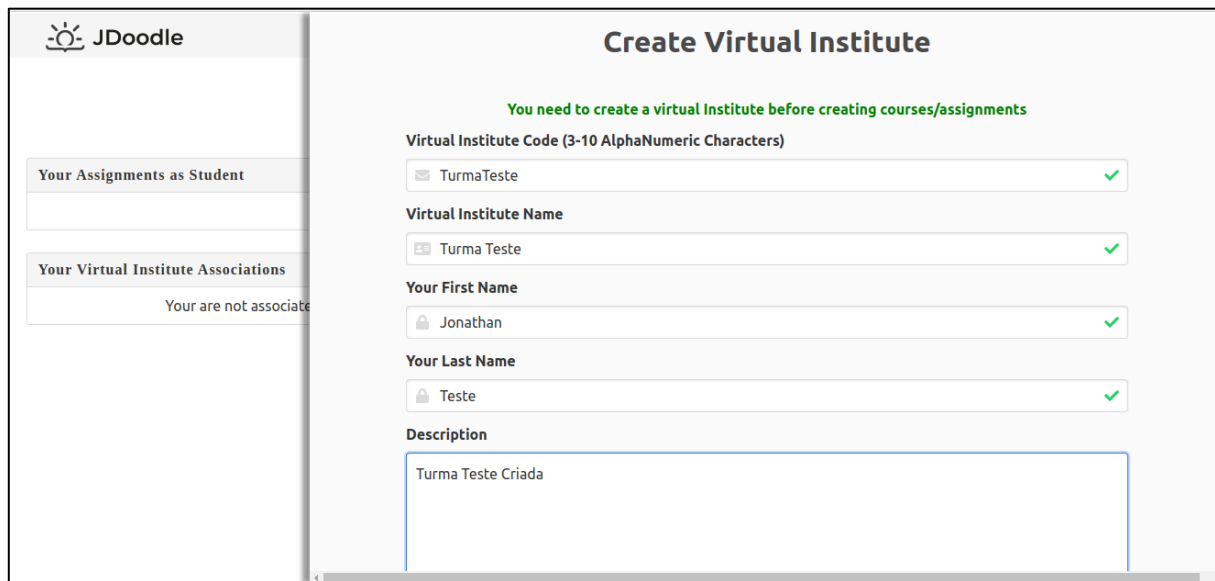
4.1 Descrição das Ferramentas *Online* Selecionadas

Essa subseção apresenta uma descrição da estrutura das três ferramentas selecionadas.

4.1.1 JDoodle

JDoodle é uma plataforma de programação *online* que oferece como recursos um compilador *online*, um ambiente textual para programação que suporta 76 linguagens e 2 Bancos de Dados, possuindo ainda ferramentas para programação colaborativa, compartilhamento de código e uma área para avaliação de códigos.

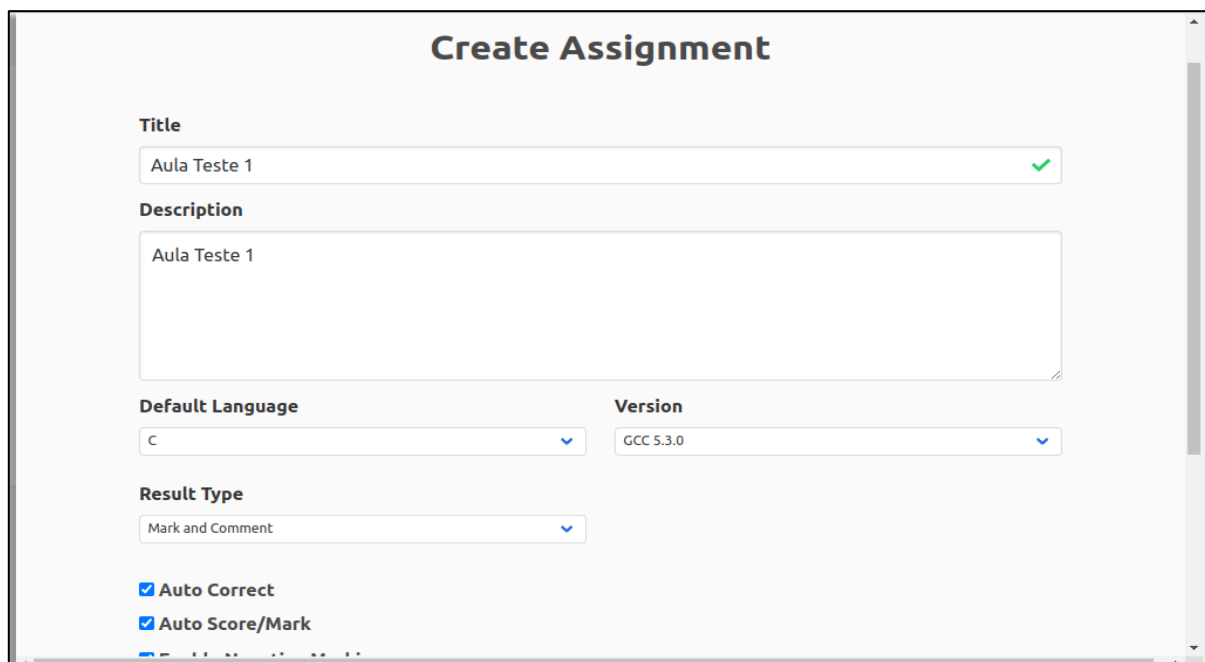
A ferramenta JDoodle oferece também recursos para criação de turmas e adição de estudantes (figura 1). Junto a esse recurso existem opções destinadas à criação de um banco de questões, tal como expresso na Figura 2, e também elementos que possibilitam a criação de atividades práticas e questionários personalizados com sistema de correção automático e/ou manual.



The screenshot shows the 'Create Virtual Institute' interface in JDoodle. On the left, there is a sidebar with navigation options: 'Your Assignments as Student', 'Your Virtual Institute Associations', and 'Your are not associate'. The main area is titled 'Create Virtual Institute' and contains a green message: 'You need to create a virtual Institute before creating courses/assignments'. Below this, there are several input fields, each with a green checkmark indicating successful validation:

- Virtual Institute Code (3-10 AlphaNumeric Characters):** TurmaTeste
- Virtual Institute Name:** Turma Teste
- Your First Name:** Jonathan
- Your Last Name:** Teste
- Description:** Turma Teste Criada

Figura 1: Criação de turma na ferramenta JDoodle. Fonte: Autoria própria.



The screenshot shows the 'Create Assignment' interface in JDoodle. The form includes the following fields and options:

- Title:** Aula Teste 1 (with a green checkmark)
- Description:** Aula Teste 1
- Default Language:** C (dropdown menu)
- Version:** GCC 5.3.0 (dropdown menu)
- Result Type:** Mark and Comment (dropdown menu)
- Auto Correct:**
- Auto Score/Mark:**

Figura 2: Criação de Banco de questões na ferramenta JDoodle. Fonte: Autoria própria.

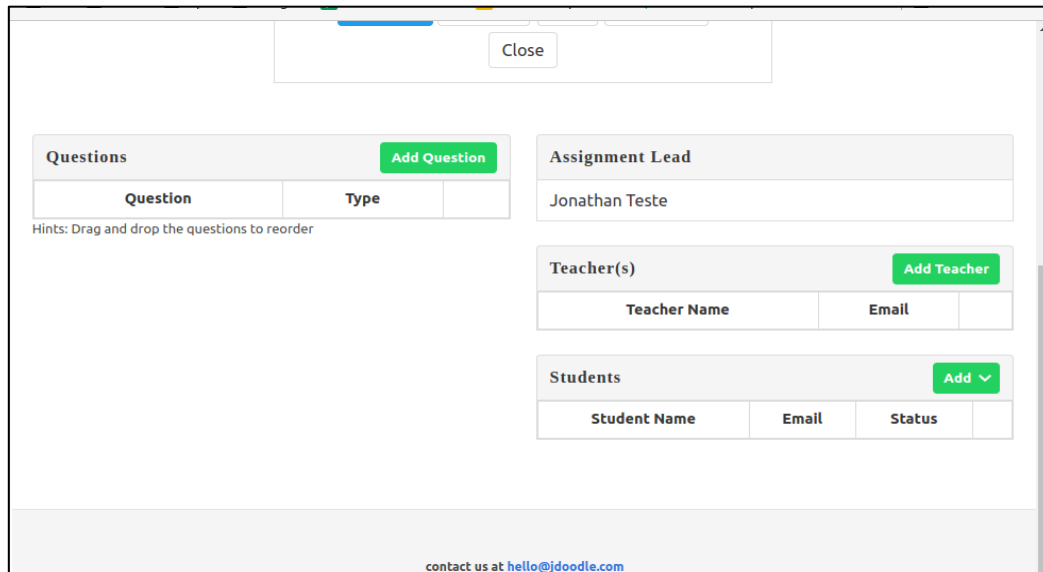


Figura 3: Definição de atividades práticas e questionários avaliativos na ferramenta JDoodle. Fonte: Autoria própria.

4.1.2 Coding Rooms

Coding Rooms é um ambiente de desenvolvimento baseado em navegador criado pela startup EdTech. A plataforma Coding Rooms oferece um plano pago e outro gratuito de operação, sendo que o plano gratuito oferece um leque de opções capaz de atender às necessidades de um professor em sala de aula.

O ambiente oferece recursos para ensino ao vivo e videoconferência, exemplificado na Figura 4, recursos para criação de turmas com instruções, monitoramento e avaliação com feedback manuais e automáticos, opção para programar tarefas com relatórios de pontuação, permite o desenvolvimento de programação textual com suporte a mais de 23 linguagens de programação, além de oferecer a opção de programação visual por blocos (fig. 5).

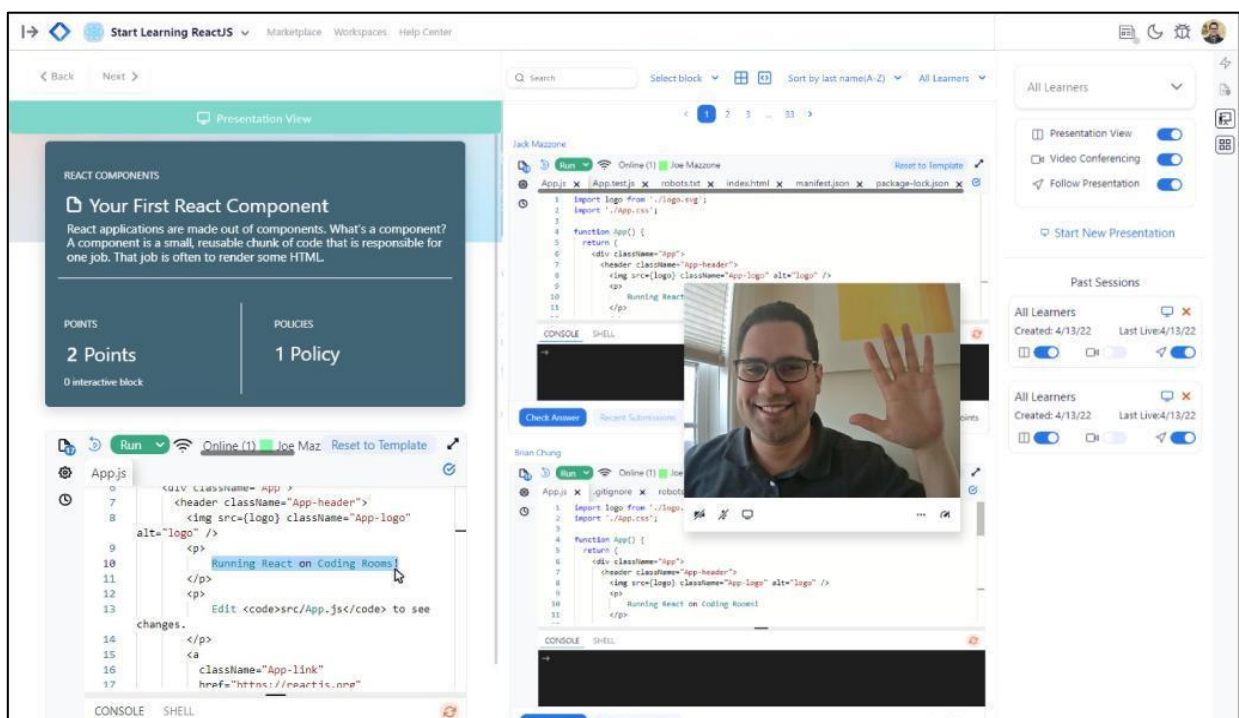


Figura 4: Aula em videoconferência na ferramenta Coding Rooms. Fonte: CODING ROOMS(2022).

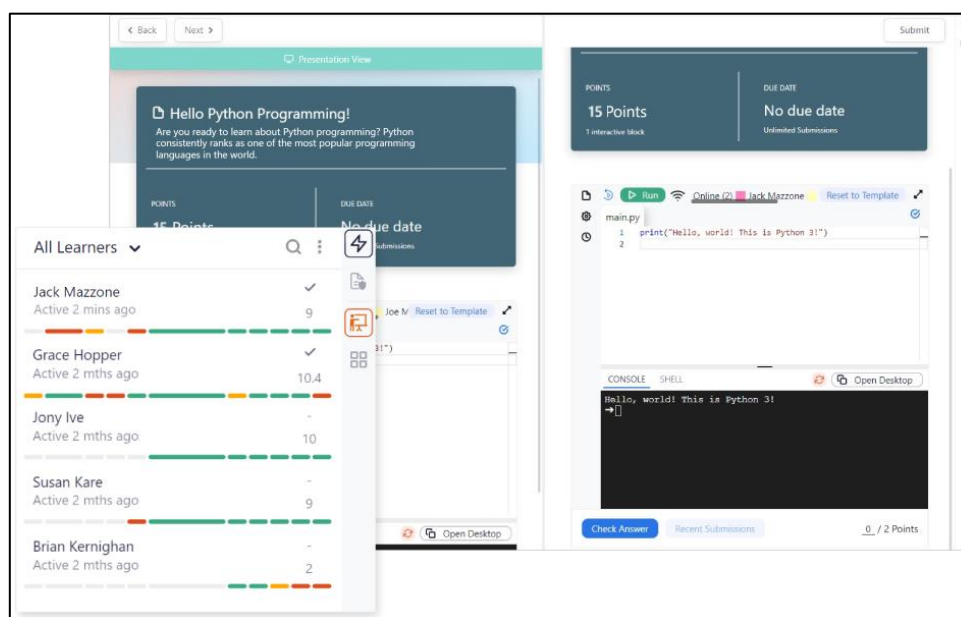


Figura 5: Aula em videoconferência na ferramenta Coding Rooms. Fonte: CODING ROOMS(2022).

Além dos recursos básicos, a ferramenta ainda fornece suporte para o desenvolvimento de aplicativos web e aplicativos desktop e ainda a possibilidade de adicionar conteúdo, interativo, vídeos e imagens na construção do material didático.

4.1.3 Online GDB

Online GDB é um ambiente de programação, que oferece na sua base ferramentas para desenvolvimento e análise de códigos de forma textual. A ferramenta suporta 25 linguagens de programação, porém a ferramenta tem uma limitação de arquivos individuais de 100 Kb⁶(KB) e projetos com múltiplos arquivos de 200 KB.

O ambiente Online GDB oferece suporte para criação e gestão de turmas com múltiplos professores, criação de banco de questões e compartilhamento de código, como exemplificado na Figura 6. O ambiente de codificação ilustrado na Figura 7 oferece um sistema de feedback textual que apresenta o resultado do processo de execução do algoritmo criado.

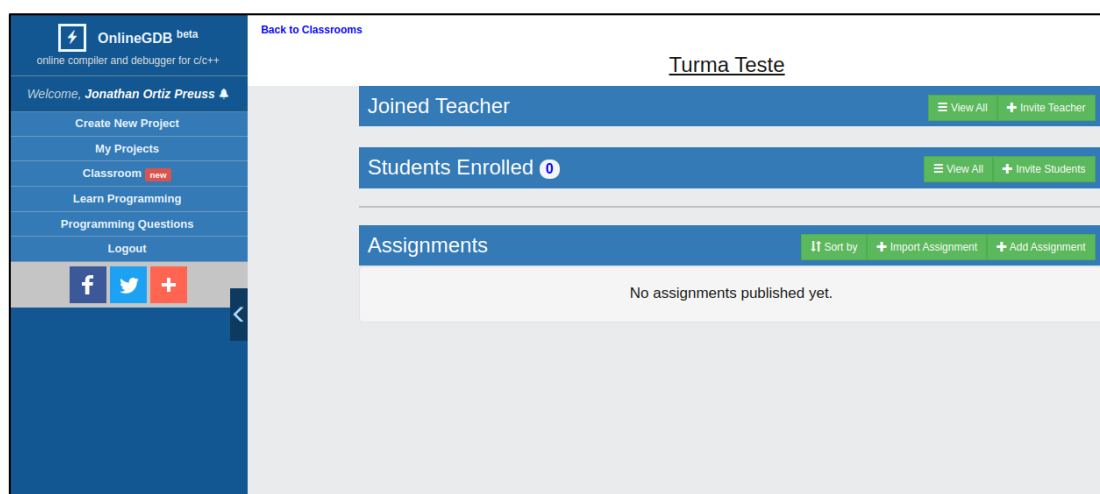


Figura 6: Definição de turmas e banco de questões na ferramenta Online GDB. Fonte: Autoria própria.

⁶ Kilobyte é uma unidade de medida utilizada na computação, formado pelo conjunto de 1024 bytes.

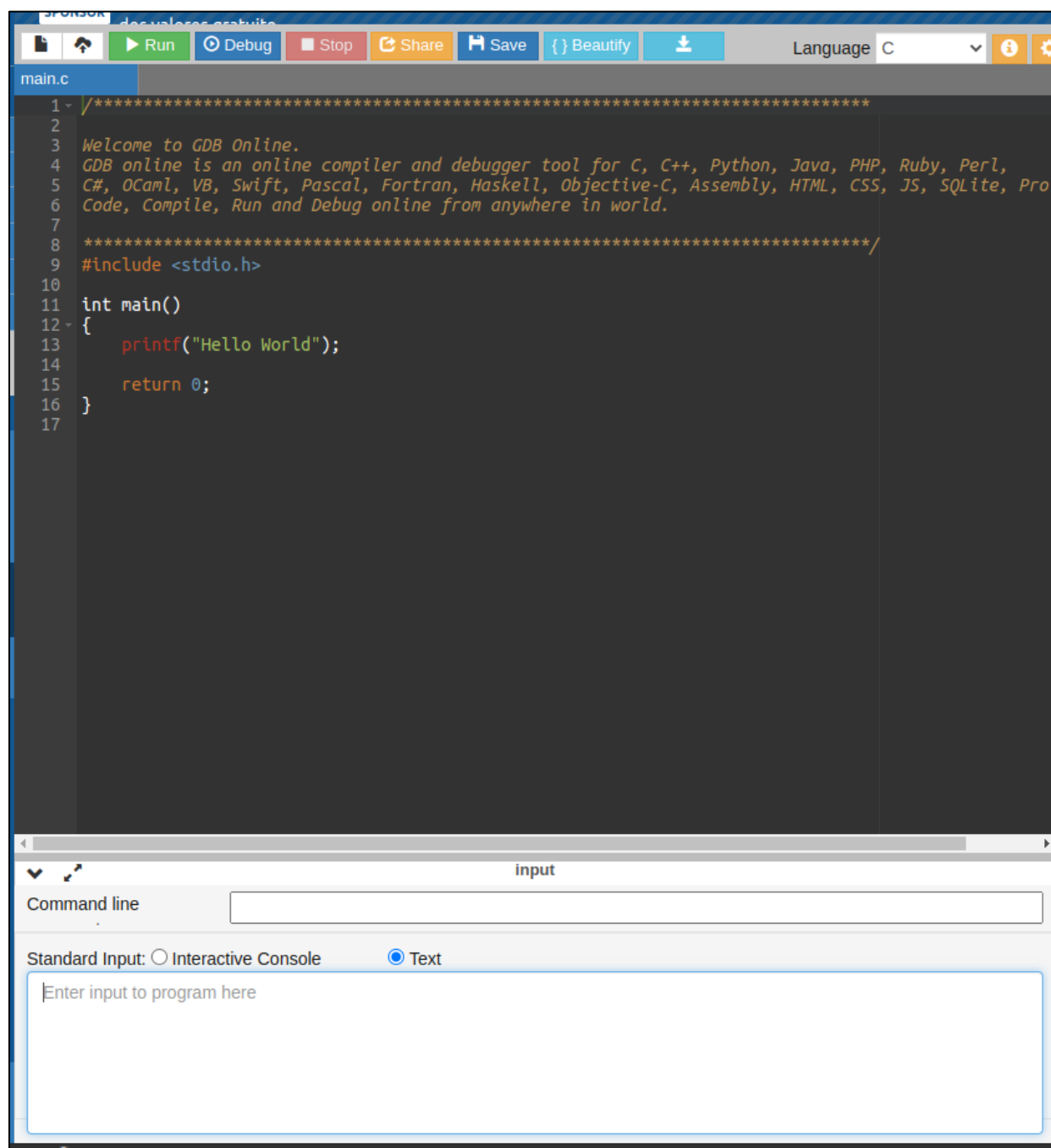


Figura 7: Ambiente de codificação na ferramenta *Online GDB*. Fonte: Autoria própria.

4.2 Descrição das Ferramentas *Online* Selecionadas

A análise de usabilidade das ferramentas *online* selecionadas foi realizada por meio de um questionário *online* respondido por alunos do curso de informática para a internet.

No contexto desse estudo, a atratividade corresponde à qualidade de uma ferramenta *online* de ser atraente ao usuário/aluno, por exemplo, o *layout*, cores e organização dos elementos na tela. A apreensibilidade é a capacidade da ferramenta analisada, de fornecer informações, diretas ou indiretas, que possibilite ao usuário aprender de forma autônoma sua aplicação. Já a inteligibilidade de uma ferramenta *online* refere-se à capacidade do usuário, após a utilização da ferramenta, conceber se a ferramenta é realmente apropriada para as tarefas as quais a mesma se propõe desempenhar. Operacionalidade é a habilidade da ferramenta de fornecer condições ao usuário operar e controlar de forma independente a ferramenta analisada.

O questionário para análise de usabilidade das ferramentas *online* selecionadas foi criado usando a plataforma Google Docs e disponibilizado para os alunos via *link* de acesso.

4.3 Avaliação das ferramentas *online*

Ao longo desta seção estão apresentados e discutidos os resultados da avaliação das ferramentas online testadas com a turma do primeiro ano do Curso Técnico em Informática para Internet Subsequente (CTISM). O grupo de alunos foi conduzido ao laboratório de informática, onde utilizou cada uma das três ferramentas por um período total de 120 minutos. O período de tempo de 120 minutos, obtido ao analisar o período de tempo entre a primeira e a última avaliação realizada pelos alunos, não foi limitado aos alunos. Após esse processo prático, os alunos foram convidados a responder o questionário com um total de 16 questões, disponibilizado via Google Docs, com os resultados obtidos expressos a seguir.

4.3.1 Avaliação da Atratividade das Ferramentas

A atratividade das ferramentas estudadas foi mensurada pela primeira questão do questionário (Considerando o layout da ferramenta, cores, fontes e a organização dos campos para escrita de código. Qual o grau de atratividade da ferramenta ?) e os resultados obtidos estão demonstrados no gráfico na Figura 8. O gráfico de colunas apresenta a porcentagem de cada opção de resposta para a primeira questão, em relação ao total de respostas dos participantes.

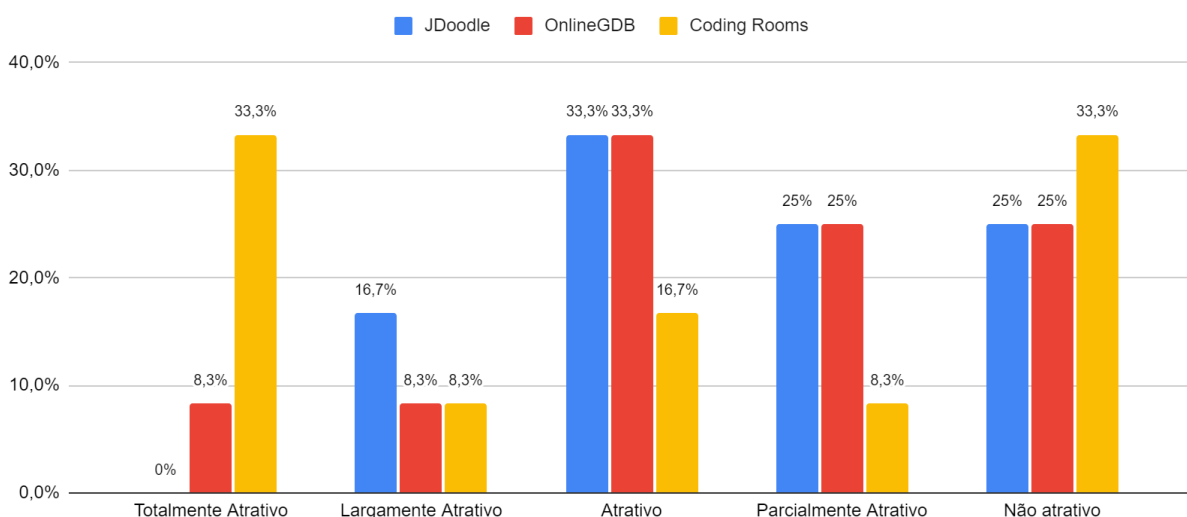


Figura 8: Gráfico de Avaliação da Atratividade. Fonte: Autoria própria.

Observa-se que a plataforma Coding Rooms tem um maior nível de atratividade dentre as ferramentas, com 33,3% dos participantes entendendo que a plataforma é Totalmente Atrativa. Ao analisar a plataforma de forma individual surge um nível de polarização na opinião do grupo de alunos, sendo 33,3% acredita que a plataforma é Totalmente Atrativa e 33,3% Não Atrativa.

As plataformas JDoodle e Online GDB apresentam também uma polarização em relação à opinião do nível de atratividade das plataformas, sendo que ambas as plataformas possuem 33,3% das respostas classificando as plataformas como Atrativas e 25% entendendo como Não Atrativas.

4.3.2 Avaliação do Grau de Inteligibilidade das Ferramentas

O grau de inteligibilidade das ferramentas estudadas, foi analisado através da segunda questão do questionário (De acordo com suas impressões sobre as ferramentas, o quanto a ferramenta é

apropriada para o auxílio no ensino de programação ?) e os resultados obtidos estão demonstrados no gráfico na Figura 9.

Cada coluna do gráfico representa a porcentagem das respostas obtidas para a segunda questão, em relação ao total de respostas dos participantes.

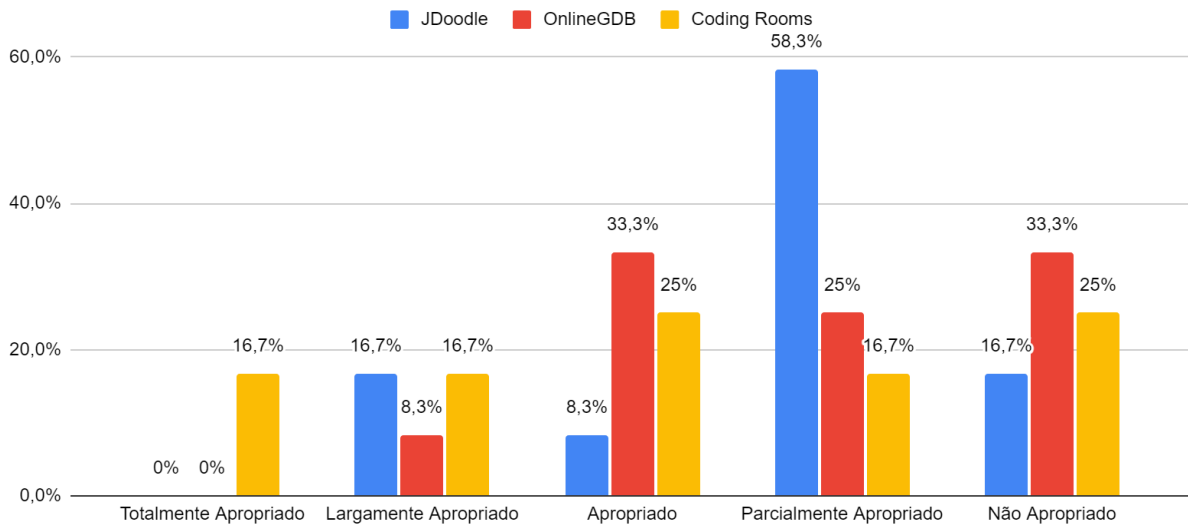


Figura 9: Gráfico de Avaliação do Grau de Inteligibilidade. Fonte: Autoria própria.

Percebe-se que a ferramenta JDoodle apresenta o maior consenso nas respostas em relação ao grau de inteligibilidade, sendo um total de 58,3% dos participantes entendendo que a ferramenta é Parcialmente Apropriada para a atividade de programação. Já a ferramenta Online GDB apresenta uma polarização nas respostas, onde 33,3% dos participantes consideram a plataforma Apropriada e 33,3% Não Apropriada.

Observa-se que os participantes não acreditam que as ferramentas oferecem grandes níveis de inteligibilidade, apenas a plataforma Coding Rooms obteve opiniões aprovando suas características de inteligibilidade, sendo que apenas 16,7% dos participantes acreditam que a plataforma é totalmente apropriada.

4.3.3 Avaliação da Apreensibilidade das Ferramentas

A apreensibilidade das ferramentas analisadas, foi analisada através da terceira questão do questionário (Considere o seu processo de interação com a ferramenta, qual o grau de adequação das funções da ferramenta que possibilite o usuário aprender como utilizar e trabalhar com a ferramenta?) e os resultados obtidos estão demonstrados no gráfico da Figura 10. Cada coluna do gráfico representa a porcentagem das respostas obtidas dos participantes do experimento em relação à apreensibilidade das ferramentas experimentadas.

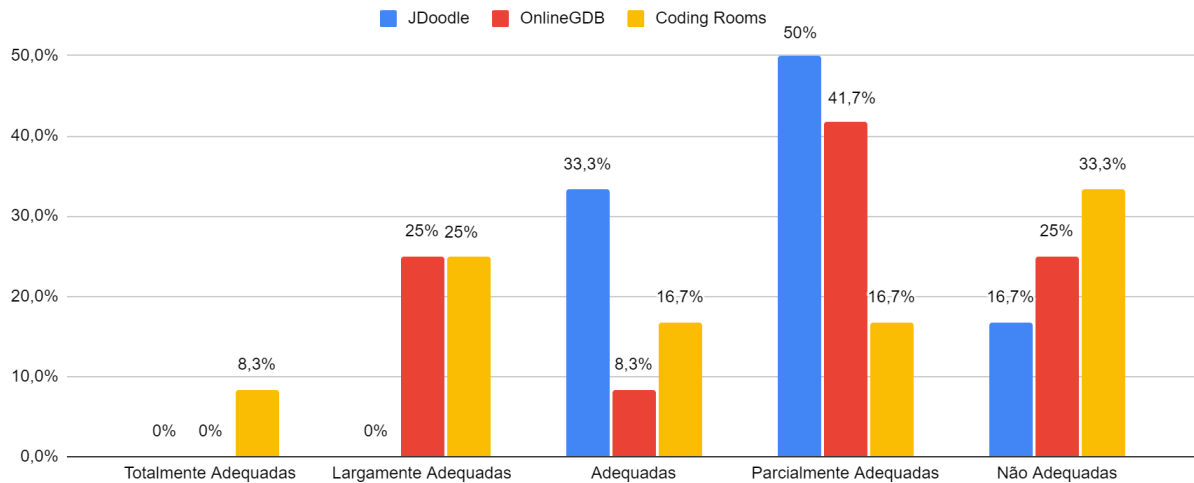


Figura 10: Gráfico de Avaliação da Apreensibilidade . Fonte: Autoria própria.

A plataforma Cooding Rooms é a única que obteve o maior nível de avaliação sobre sua apreensibilidade, com 8,3% das respostas demonstrando que os usuários acreditam que a plataforma possui elementos totalmente adequados para sua apreensibilidade. O nível de avaliação de apreensibilidade das ferramentas se concentrou entre os níveis de apreensibilidade Parcialmente Adequada e Não Adequada. De acordo com as respostas 50,0% dos participantes entendem que JDoodle é parcialmente adequada, já 41,7% acreditam que Online GDB é parcialmente adequada e por fim 16,7% apontam Coding Rooms como uma ferramenta parcialmente adequada.

A porcentagem das respostas que avaliam a apreensibilidade das ferramentas como não adequada, é proporcionalmente opostas a aquelas que avaliam como parcialmente adequadas, sendo JDoodle como 16,7%, Online GDB com 25,0% e Coding Rooms com 33,3% das respectivas a porcentagem de avaliações das ferramentas como “não adequadas”.

4.3.4 Avaliação da Operacionalidade das Ferramentas

A quarta questão do questionário foi utilizada para obter a opinião dos participantes sobre o nível de operacionalidade das ferramentas analisadas, os resultados obtidos estão elucidados no gráfico na Figura 11. Cada coluna do gráfico representa a porcentagem das respostas obtidas dos participantes sobre cada uma das ferramentas experimentadas.

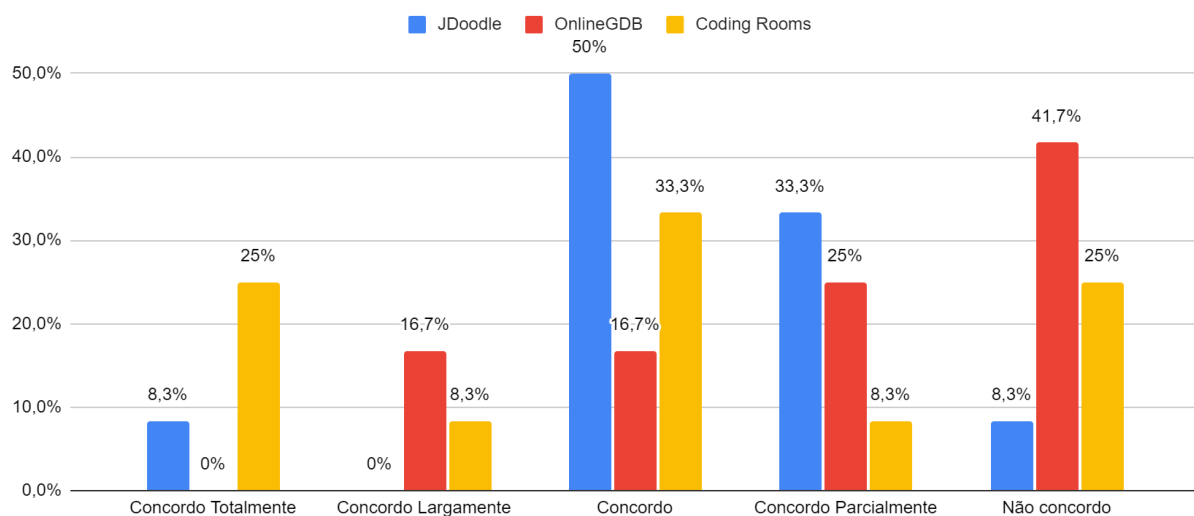


Figura 11: Gráfico de Avaliação da Operacionalidade. Fonte: Autoria própria.

De acordo com as respostas, 50,0% dos participantes concordam que as características da ferramenta JDoodle oferecem operacionalidade para a proposta de programação. Porém observa-se que existe uma baixa porcentagem de participantes que concordam totalmente que os mecanismos das ferramentas oferecem um nível alto de operacionalidade, sendo que apenas 8,3% da plataforma JDoodle e 25,0% para plataforma coding Roms dos participantes concordam totalmente com a operacionalidade dessas ferramentas.

4.3.5 Avaliação do Nível de Aceitação, Recorrência e Recomendação das Ferramentas

Por fim, o último quesito avaliado foi a relação de aceitação e recomendação das ferramentas. após a experiência de uso dos participantes do experimento, os resultados obtidos para essa questão estão apresentados no gráfico na Figura 12. Observa-se que existe uma grande rejeição da utilização das ferramentas estudadas. As respostas obtidas indicam que 66,7% dos participantes não usariam e não recomendariam as ferramentas JDoodle e Online GDB apresentam.

Apesar da rejeição de 50% em sua adoção e recomendação, a plataforma Coding Rooms é a plataforma que tem o maior percentual entres os usuários que demonstraram interesse em continuar utilizando a plataforma, sendo que 33,3% dos participantes voltariam a usar e recomendar fortemente a Coding Rooms.

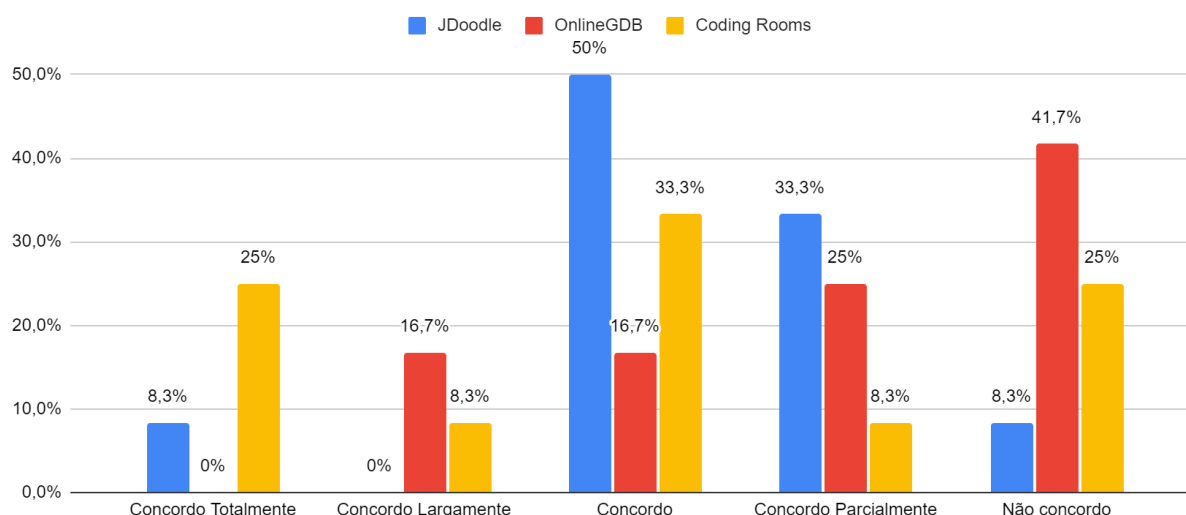


Figura 12: Gráfico de Avaliação da Aceitação e Recomendação. Fonte: Autoria própria.

4.3.6 Análise Geral dos Resultados Obtidos

Ambas as ferramentas analisadas possuem pontos fortes e atrativos técnicos que as tornam adequadas para o ensino de programação no contexto remoto, tais como um ambiente textual para programação de múltiplas linguagens, mecanismos para programação colaborativa, criação e gestão de turmas, compartilhamento e avaliação de códigos e acompanhamento do progresso dos alunos. A plataforma Coding Rooms tem um ponto extra, diferente das outras, que é oferecer, de forma nativa, ferramentas para videoconferências.

Em relação às limitações, pode-se apontar que a plataforma Online GBD apresenta uma limitação significativa, que é a restrição do tamanho de arquivos. Já as plataformas JDoodle e a Coding Rooms possuem planos pagos, porém os recursos disponibilizados gratuitamente são suficientes para o desenvolvimento do ensino de programação.

Ao fim da análise dos resultados do questionário avaliativo submetido aos alunos, observa-se que não existe uma plataforma que obteve unanimidade nos quesitos de usabilidade. Com o objetivo de entender melhor os resultados obtidos, a Tabela 7 apresenta a relação das ferramentas

com maior porcentagem e os itens com maior peso avaliativo de cada uma das questões do questionário de usabilidade.

Tabela 7: Relação de Ferramentas Com Maior Avaliação Por Questão.

Item avaliativo	Ferramenta		
	JDoodle	Online GDB	Coding Rooms
Totalmente Atrativo			X
Totalmente Adequado			X
Totalmente Adequadas			X
Concordo Totalmente			X
Usaria e Recomendaria Fortemente			X

Fonte: Autoria própria.

Mesmo os itens avaliativos de maior importância do questionário recebendo poucas respostas, entre as respostas analisadas e a análise das características técnicas observou-se que Coding Rooms é a plataforma que apresenta mais potencialidades técnicas e contempla os cinco itens de maior peso avaliativo sob a perspectiva dos alunos participantes do experimento.

5 Considerações Finais

O ensino de programação é rodeado de dificuldades e peculiaridades e o ensino remoto adicionou uma série de novas dificuldades anteriormente pouco discutidas, como a falta de recursos para os estudantes e professores, falta de ferramentas para o cumprimento das atividades pedagógicas síncronas ou assíncronas de forma *online*, dentre outras. Diante do exposto, o objetivo deste trabalho foi a identificação, teste e análise das plataformas e ferramentas gratuitas *online* que podem auxiliar os discentes e docentes a contornar as adversidades enfrentadas no processo de ensino e aprendizagem das disciplinas de algoritmos e programação no contexto do ensino remoto.

Após realizar uma análise dos trabalhos presentes na literatura para a identificar as principais dificuldades que permeiam o ensino de programação e as novas barreiras que surgiram no ensino remoto, selecionaram-se ferramentas com características que podem auxiliar as atividades de ensino, executando um experimento com um grupo de alunos para mensurar o nível de usabilidade e aceitação das ferramentas testadas, discutindo e analisando os dados obtidos observa-se que a plataforma Coding Room se destaca obtendo uma elevada pontuação na análise de usabilidade e com o maior número de potencialidades técnicas, oferecendo de forma gratuita ferramentas necessárias para o ensino remoto.

A primeira barreira encontrada no desenvolvimento deste trabalho foi a pequena amostragem de alunos que participaram do experimento. Apesar de ser um ambiente parcialmente controlado, não havia garantias de que todos os alunos que responderam o questionário interagiram com as ferramentas propostas. O tempo de experimentação dos alunos com cada uma das ferramentas pode ser considerado um ponto limitante do trabalho, pois houve pouco tempo para que os alunos se familiarizassem com o funcionamento de cada ferramenta, não sendo possível assegurar que os alunos utilizaram todos os recursos de cada ferramenta.

Para os trabalhos futuros propõe-se um estudo específico sobre a plataforma Coding Room, aplicando experimentos para analisar a experiência de usabilidade da mesma em dispositivos móveis como celulares e *tablets*. Propõe-se, ainda, a execução do processo de análise de usabilidade das ferramentas com um maior grupo de alunos e, adicionalmente, de professores de todos os níveis de ensino, visando coletar e analisar o ponto de vista de toda população-alvo deste estudo.

Por fim, espera-se que este trabalho contribua, através da metodologia e resultados apresentados, para o auxílio dos professores na escolha de uma ferramenta *online* que possa

contribuir para o processo de ensino e aprendizagem das disciplinas de Programação e Algoritmos.

Edição Especial: Metodologias de ensino e ferramentas tecnológicas de suporte para o ensino remoto no Pós-Pandemia

Esta publicação compõe a edição especial “Metodologias de ensino e ferramentas tecnológicas de suporte para o ensino remoto no Pós-Pandemia”, conduzida pelo Editor convidado Prof. Dr. Marciel Aparecido Consani (Universidade de São Paulo).

Referências

- Arimoto, M. M., & Oliveira, T. D. O. (2019). Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação. Em *XXVII Workshop sobre Educação em Computação*, 27 (pp. 244-254), Belém. <https://doi.org/10.5753/wei.2019.6633> [GS Search]
- Bennedsen, J. & Caspersen, M. E. (2019). Failure rates in introductory programming: 12 years later. Em *ACM Inroads*: Vol. 10(2). (pp. 30-36). New York, EUA: ACM. <https://doi.org/10.1145/3324888> [GS Search]
- Bosse, Y. & Gerosa, M. A. (2016). SHARE ON Why is programming so difficult to learn?: Patterns of Difficulties Related to Programming Learning Mid-Stage. Em *ACM SIGSOFT Software Engineering Notes*: Vol. 41(6). (pp. 1-6). New York, EUA: ACM. <https://doi.org/10.1145/3011286.3011301> [GS Search]
- Bosse, Y. & Gerosa, M. A. (2017). Difficulties of Programming Learning from the Point of View of Students and Instructors. Em *IEEE Latin America Transactions*: Vol. 15(11). (pp. 2191-2199). IEEE. <https://doi.org/10.1109/TLA.2017.8070426> [GS Search]
- Berssanette, J. H. & Francisco, A. C. D. (2021). Formação Pedagógica e o Ensino-Aprendizagem de Programação: Um estudo preliminar. Em *INFORMÁTICA NA EDUCAÇÃO teoria & prática*: Vol. 24(1). (pp. 1-6). Porto Alegre, Brasil. <https://doi.org/10.22456/1982-1654.108240> [GS Search]
- Brasil. (2020, Agosto 18). Lei Nº 14.040, Presidência da República. Secretaria-Geral Subchefia para Assuntos Jurídicos. Recuperado em 10 jun. 2022. de http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/lei/L14040.htm
- Cheah, C. S. (2020). Factors Contributing to the Difficulties in Teaching and Learning of Computer Programming: A Literature Review. Em *IEEE Frontiers in Education Conference (FIE) Proceedings*: Vol. 12(2). (pp. ep272). Londres, UK: <https://doi.org/10.30935/cedtech/8247> [GS Search]
- Gomes, A. & Mendes, A. (2014). A teacher's view about introductory programming teaching and learning: Difficulties, strategies and motivations. Em *IEEE Frontiers in Education Conference (FIE) Proceedings*. (pp. 1-8). Madrid, Espanha,: IEEE. <https://doi.org/10.1109/FIE.2014.7044086> [GS Search]
- Holanda, W. D. D., Freire, L. D.P. & Coutinho, J. C. D. S. (2019). Estratégias de ensino-aprendizagem de programação introdutória no ensino superior: uma Revisão Sistemática da Literatura. Em *RENOTE*: Vol. 17(2). (pp. 527-536). Porto Alegre (RS). <https://doi.org/10.22456/1679-1916.95905> [GS Search]

- Hussain, A. J., Fergus, P., Al-Jumeily, D., Pich & A., Hind, J. (2015). Teaching Primary School Children the Concept of Computer Programming. Em *Anais IEE International Conference On Developments In Esystems Engineering (Dese)*. Burj Khalifa, Dubai: IEEE. <https://doi.org/10.1109/DeSE.2015.57> [GS Search]
- ISO/IEC 25010. (2020). Site Norma ISO25000. Recuperado em 10 jun. 2022. de <https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>
- IBGE (2021, Agosto 27). Estimativas da população residente no Brasil e Unidades da Federação com data de referência em 1º de julho de 2021. Recuperado em 10 jun. 2022. de https://ftp.ibge.gov.br/Estimativas_de_Populacao/Estimativas_2021/estimativa_dou_2021.pdf
- Joye, C. R., Moreira, M. M. & Rocha, S. S. D. (2020). Distance Education or Emergency Remote Educational Activity: in search of the missing link of school education in times of COVID-19. Em *Research, Society and Development Journal*: Vol. 9(7). (pp. e521974299). Vargem Grande Paulista (SP). <https://doi.org/10.33448/rsd-v9i7.4299> [GS Search]
- Junior, S. M. D. S. & Morais, M. A. C. D. (2020). Mapping Of Tools Used In Teaching Programming Em *Brazilian Journal of Development (BJD)*: Vol. 6(11). (pp. 87466-87476). Curitiba (PR). <https://doi.org/10.34117/bjdv6n11-243> [GS Search]
- Júnior, D. L. G. & Boga, D. T. R. (2020). Ensino De Programação: Uma Revisão Sistemática E as Aplicações Ao Ensino Profissional. Em *Cadernos da FUCAMP*: Vol. 19(41). (pp. 14-30). Monte Carmelo (MG). Recuperado em <https://brazilianjournals.com/ojs/index.php/BRJD/article/view/19769> [GS Search]
- Kadar, R., Wahab, N. A., Othman, J. & Shamsuddin, M., Mahlan, S. B. (2016). A Study of Difficulties in Teaching and Learning Programming: A Systematic Literature Review. Em *International Journal of Academic Research in Progressive Education and Development*: Vol. 10(3). (pp. 591-605). New Satelite Towno (PAK). <https://doi.org/10.6007/IJARPED/v10-i3/11100> [GS Search]
- Khouri, C. M. B., Santos, G. N. D. & Barbosa, M. S. S. (2020). Mapeamento Sistemático em Metodologias de Ensino-aprendizagem de Programação. Em *Revista de Ciência da Computação (ReCiC)*: Vol. 2(1). (pp. 13-27). Vitória da Conquista (BA). <https://doi.org/10.22481/recic.v2i1.6669> [GS Search]
- Law, K. M. Y., Lee, V. C. S. & Yu, Y. T. (2010). Learning motivation in e-learning facilitated computer programming courses. Em *Computers & Education Journal*: Vol. 55. (pp. 218-228). <https://doi.org/10.1016/j.compedu.2010.01.007> [GS Search]
- Lima, C. C. (2018). Os Desafios da Educação superior: a formação docente inicial para a utilização pedagógica das tecnologias digitais. Em EIDELWEIN, M. P., GOMES, R. S. (Org.), *Circulando entre as possibilidades de educação à distância e os desafios da educação superior led*. Porto Alegre (RS): Editora Cirkula.
- Lima, C. C. Mattar, J. Utilização do design educacional na concepção do projeto de ensino de programação de computadores na modalidade EaD. *Research, Society and Development*, v. 4, n. 3, p. 199-214, 2017. <https://doi.org/10.17648/rsd-v4i3.84> [GS Search]
- Morais, C. G. B., Neto, Fr. M. M. & Osório, A. J. M. (2020). Difficulties and challenges in the learning process of algorithms and programming in higher education: a systematic literature review. Em *Research, Society And Development*,: Vol. 9(10). (pp. e9429109287). Vargem Grande Paulista (SP). <https://doi.org/10.33448/rsd-v9i10.9287> [GS Search]

- Medeiros, R. P., Ramalho, G. L. & Falcão, T. P. (2018). A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education. Em *IEEE Transactions on Education*: Vol. 62(2). (pp. 77-90). Londres (UK): IEEE. <https://doi.org/10.1109/TE.2018.2864133> [GS Search]
- Medina, M. & Ferting, C. (2005). *Algoritmos e Programação: Teoria e Prática*. 2. ed. São Paulo: Novatec Editora Ltda.
- Qian, Y. & James, L. (2017). Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review. Em *Transactions on Computing Education*: Vol. 18(1). (pp. 1-24). New York, EUA: ACM. <https://doi.org/10.1145/3077618> [GS Search]
- Rodrigues, S. M., Venero, M. F., Rodriguez, C., Goya, D. & Rocha, R. V. D. (2019). Avaliando ambientes para ensino de programação com suporte para o desenvolvimento da metacognição. Em *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*. (pp. 417-427). Brasília (DF). <http://dx.doi.org/10.5753/cbie.wcbie.2019.417> [GS Search]
- Santos, S. C. D. S., Santana, E., Santana, L., Rossi, P., Cardoso, L., Fernandes, U., Carvalho, C. & Tôrres, P. (2018). Applying PBL in Teaching Programming: an Experience Report. Em *IEEE Frontiers in Education Conference (FIE)*. San Jose (CA): IEEE. <https://doi.org/10.1109/FIE.2018.8658978> [GS Search]
- Simon, Luxton-Reilly, A., Ajanovski, V. V., Fouh, E., Gonsalvez, C., Leinonen, J., Parkinson, J., Poole M. & Thota, N. (2019). Pass Rates in Introductory Programming and in other STEM Disciplines. Em *ITiCSE-WGR '19: Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education*: Vol. 15(11). (pp. 53-71). Aberdeen Scotland (UK): IEEE. <https://doi.org/10.1145/3344429.3372502> [GS Search]
- Souza, D. M., Batista, M. H. D. S. & Barbosa, E. F. (2016). Problemas e Dificuldades no Ensino de Programação: Um Mapeamento Sistemático. Em *Brazilian Journal of Computers in Education*: Vol. 24(1). (pp. 39-52). Porto Alegre (RS). <http://dx.doi.org/10.5753/rbie.2016.24.1.39> [GS Search]
- Shute, V. J., Sun, C. & Asbell-clarke, J. (2017). Demystifying computational thinking. Em *Educational Research Review*: Vol. 22. (pp. 142-158). <https://doi.org/10.1016/j.edurev.2017.09.003> [GS Search]
- Wainer, J. (2007, Maio 27). Métodos de pesquisa quantitativa e qualitativa para a Ciência da Computação. Recuperado em 30 jun. 2022. de http://www.planalto.gov.br/ccivil_03/_ato2019-2022/2020/lei/L14040.htm