

## Juiz Online no ensino de Programação Introdutória - Uma Revisão Sistemática da Literatura

Rodrigo Elias Francisco<sup>1</sup>, Cleon X. Pereira Júnior<sup>2</sup>, Ana Paula Ambrósio<sup>3</sup>

<sup>1</sup>Instituto Federal Goiano (IFGoiano)  
Morrinhos – GO – Brasil

<sup>2</sup>Instituto Federal Goiano (IFGoiano)  
Campos Belos – GO – Brasil

<sup>3</sup>Instituto de Informática – Universidade Federal de Goiás (UFG)  
Goiânia – GO – Brasil

{rodrigo.francisco, cleon.junior}@ifgoiano.edu.br, apaula@inf.ufg.br

**Abstract.** *The use of Online Judge systems in computer education can automate student feedback and make teacher work more productive. This article presents a Systematic Review of the Literature (RSL) that contributes to the specification of an Online Judge for use in introductory programming education (CS1). The RSL synthesis is subdivided into: benefits, problems, functional requirements and non-functional requirements. The results indicate aspects that must be taken into account when choosing an Online Judge for CS1, as well as a list of tools and how they treat these aspects. Critical issues related to the validation of tools were observed.*

**Resumo.** *O uso de sistemas de Juiz Online no ensino de computação pode automatizar o feedback oferecido ao aluno e tornar o trabalho do professor mais produtivo. Este artigo apresenta uma Revisão Sistemática da Literatura (RSL) que contribui para a especificação de um Juiz Online a ser usado no ensino de programação introdutória (CS1). Para isso, a síntese da RSL se subdividiu em: benefícios, problemas, requisitos funcionais e requisitos não-funcionais. Os resultados apontam para aspectos que devem ser levados em consideração na escolha de um Juiz Online para CS1, assim como uma lista de ferramentas e como estas tratam os aspectos levantados. Observou-se que a seleção das ferramentas ainda apresenta questões críticas relacionadas à validação das mesmas.*

### 1. Introdução

As disciplinas de Introdução à Programação de Computadores, conhecidas na literatura como *Computer Science 1* (CS1), são objetos de pesquisas científicas dada a dificuldade que muitos alunos sentem na elaboração de algoritmos e implementação dos mesmos usando uma linguagem de programação. Programar exige elevado nível de generalização, abstração, pensamento crítico [Gomes et al. 2008] e a presença de modelos mentais adequados [Byrne and Lyons 2001, Wilson and Shrock 2001]. Outros fatores, como motivação [Rountree et al. 2004], também têm impacto na aprendizagem de CS1.

Apesar de não haver um consenso quanto à melhor metodologia ou linguagem de programação a ser usada, observa-se que a disciplina de CS1 necessita de que o aluno resolva uma grande quantidade de problemas. No entanto, em uma turma heterogênea, o planejamento e correção de problemas gera uma sobrecarga no professor, dificultando o *feedback* aos alunos, o que pode causar frustrações no processo de ensino-aprendizagem

[Vier et al. 2015]. Ferramentas que auxiliem neste processo têm sido propostas para preencher essa lacuna, sendo que uma das estratégias mais promissoras são os Juízes *Online*.

Um Juiz Online, também chamado de sistema de avaliação automática, oferece problemas para serem resolvidos com a submissão de códigos-fonte, em uma linguagem de programação, e os corrigem automaticamente [Kurnia et al. 2001] usando casos de teste. Cada caso de teste possui um conjunto de entradas e suas respectivas saídas. Para verificar se a resposta de um problema é uma solução correta, um programa passa por seu respectivo conjunto de testes. Esses sistemas são muito utilizados em competições de programação, como a Maratona de Programação organizada pela Sociedade Brasileira de Computação (SBC), e vêm sendo gradativamente adaptados para o ensino de CS1 devido à funcionalidade de automatizarem a correção dos programas.

Este artigo visa apresentar uma RSL sobre o uso de sistemas de Juiz *Online* no ensino de CS1, buscando contribuições à pergunta "Qual é a especificação mais adequada para que um juiz online atenda à disciplina de CS1?". Houve um planejamento com foco em mostrar os benefícios, os problemas, os requisitos funcionais e os requisitos não-funcionais sintetizados a partir dos trabalhos lidos, pois entende-se que diferentes contextos de ensino podem exigir diferentes estratégias educacionais.

## 2. Metodologia da revisão

Esta RSL segue as instruções elaboradas por [Kitchenham 2004], cuja condução se subdivide-se em três etapas: Planejamento, Execução e Resultados (Figura 1).

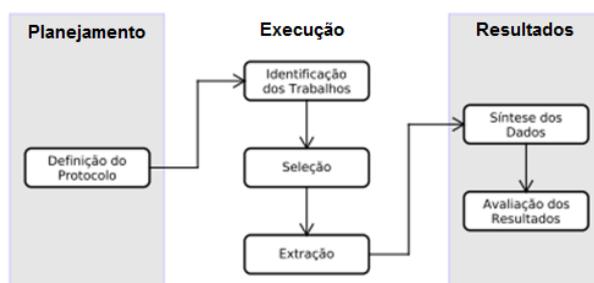


Figura 1. Fases de uma RSL - adaptado de [Vieira 2016]

### 2.1. Planejamento

Esta pesquisa tem como objetivo para responder à questão: "Qual é a especificação mais adequada para que um Juiz Online atenda à disciplina de CS1?". Para isso, outras quatro questões de pesquisa foram definidas:

- **QP1:** Quais os benefícios ao usar Juiz *Online* no ensino de CS1?
- **QP2:** Quais os problemas ao usar Juiz *Online* no ensino de CS1?
- **QP3:** Quais requisitos funcionais um Juiz *Online* precisa atender para ser usado no ensino de CS1?
- **QP4:** Quais requisitos não-funcionais um Juiz *Online* precisa atender para ser usado no ensino de CS1?

Para responder às perguntas levantadas, foi realizada uma busca nas bases de artigos científicos, considerando os trabalhos de 2010 a 2016 (Com a data limite de

14/02/2016). Foram selecionadas as bases de pesquisa: *IEEE Explorer Digital Library*, *ACM Digital Library*, *Springer*, *Science Direct* e *Scopus*, que disponibilizam mecanismo de consulta via web, estão relacionadas a temas de Computação e Informática e também permitem filtro por ano de publicação. Foram considerados trabalhos publicados em português e inglês.

**Tabela 1. Strings de busca para a seleção de artigos**

STRINGS DE BUSCA	
Idioma	String
Português	("juiz online"OR "avaliação automática"OR "correção automática") AND ("programação introdutória"OR "introdução a programação"OR "ciência da computação 1"OR "CS1")
Inglês	("online judge"OR "automated assessment"OR "automatic assessment") AND ("introductory programming"OR "introduction to programming",OR "computer science 1",OR "CS1")

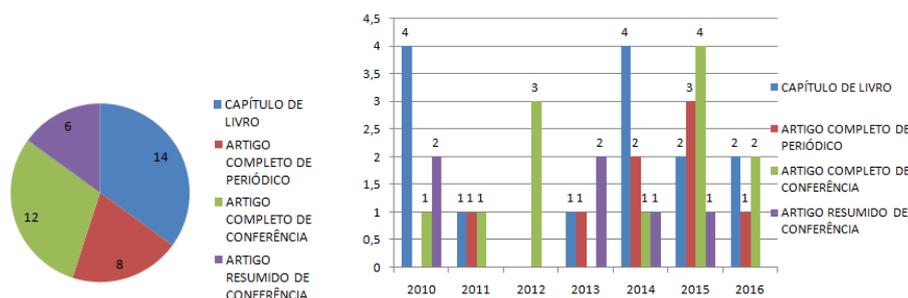
A tabela 2 apresenta os critérios de inclusão e exclusão. Como a pesquisa tem foco em abordar *Juiz Online* no ensino de CS1, os critérios de inclusão foram baseados em benefícios, problemas e os requisitos que devem ter um *Juiz Online* para ser utilizado no ensino de CS1.

**Tabela 2. Critérios de Inclusão e Exclusão**

Critérios	ID	Descrição
Exclusão	E1	Texto completo não disponível para acesso na Web
	E2	Escrito em outro idioma
	E3	Trabalho não aborda <i>Juiz Online</i> e ensino de CS1
	E4	Não é um capítulo de livro com resumo ou artigo de periódico ou de conferência
Inclusão	I1	Aborda os benefícios ao usar <i>Juiz Online</i> no ensino de CS1
	I2	Aborda os problemas ao usar <i>Juiz Online</i> no ensino de CS1
	I3	Aborda os requisitos funcionais que um <i>Juiz Online</i> precisa atender para ser usado no ensino de CS1
	I4	Aborda os requisitos não-funcionais que um <i>Juiz Online</i> precisa atender para ser usado no ensino de CS1

## 2.2. Execução

Um total de 40 trabalhos foram identificados, que seguem distribuições por ano e tipo de publicação conforme a Figura 2.



**Figura 2. Número de Trabalhos identificados agrupados por tipo de publicação e por ano e tipo de publicação**

Os dados mostram que as pesquisas sobre o assunto estão se popularizando. 2015 foi o ano em que houve o maior número de publicações, totalizando 10. Durante os dois primeiros meses de 2016, incluídos na pesquisa, de 01/01/2016 até 14/02/2016, ocorreram 5 publicações sobre o assunto. Isso indica um aumento no interesse pelo assunto a partir de 2015.

A seleção preliminar dos trabalhos contou com a leitura do título, palavras-chave e resumo de cada trabalho. Dos artigos recuperados, 26 foram excluídos por não estarem ligados ao tema (E3) e 2 por não estarem disponíveis (E1). Os 12 trabalhos restantes passaram pela fase de extração, e foram usados para responder as questões de pesquisa, através da leitura completa dos trabalhos (Tabela 3).

**Tabela 3. Trabalhos incluídos na extração conforme critérios de inclusão**

CRITÉRIO	REFERÊNCIAS	Nº TRABALHOS
I1	[Wang et al. 2011] [AlShamsi and Elnagar 2011] [Pettit et al. 2015b] [Rajala et al. 2016] [Llana et al. 2012] [Georgouli and Guerreiro 2010] [Petit et al. 2012] [Souza et al. 2015] [Vihavainen et al. 2013] [Sun et al. 2014]	10
I2	[Wang et al. 2011] [AlShamsi and Elnagar 2011] [Pettit et al. 2015a] [Pettit et al. 2015b] [Llana et al. 2012] [Georgouli and Guerreiro 2010] [Pozenel et al. 2015] [Petit et al. 2012] [Souza et al. 2015] [Vihavainen et al. 2013] [Sun et al. 2014]	11
I3	[Wang et al. 2011] [AlShamsi and Elnagar 2011] [Pettit et al. 2015a] [Rajala et al. 2016] [Llana et al. 2012] [Georgouli and Guerreiro 2010] [Pozenel et al. 2015] [Petit et al. 2012] [Souza et al. 2015] [Vihavainen et al. 2013] [Sun et al. 2014]	11
I4	[Rajala et al. 2016] [Llana et al. 2012] [Georgouli and Guerreiro 2010] [Pozenel et al. 2015] [Petit et al. 2012] [Vihavainen et al. 2013] [Sun et al. 2014]	7

### 3. Resultados

#### QP1: Quais os benefícios ao usar *Juiz Online* no ensino de CS1?

A Tabela 4 apresenta um resumo dos benefícios encontrados nos artigos selecionados. Existe um consenso de que o uso das ferramentas ajuda no ensino de CS1 por oferecer um ambiente que permite ao aluno regular sua aprendizagem, aumentando sua motivação, e, ao mesmo tempo, reduz a carga de trabalho do professor, que tem a sua disposição um conjunto de problemas já elaborados e testados e a possibilidade de automatizar a detecção de plágio e a emissão de relatórios.

**Tabela 4. Benefícios encontrados nos trabalhos**

BENEFÍCIO	REFERÊNCIAS	Nº TRABALHOS
A - Existência de sistemas que são úteis para o ensino	[Pettit et al. 2015b], [Souza et al. 2015] [Vihavainen et al. 2013], [Petit et al. 2012], [Llana et al. 2012], [Sun et al. 2014]	6
B - Redução da carga de trabalho do professor	[Rajala et al. 2016], [Wang et al. 2011], [AlShamsi and Elnagar 2011]	3
C - Oferecer lista de exercícios extensa	[Petit et al. 2012]	1
D - Preocupações pedagógicas na criação dos problemas	[Llana et al. 2012], [Sun et al. 2014], [Souza et al. 2015]	3
E - Reduzir a ocorrência de plágio	[Rajala et al. 2016]	1
F - Aprendizagem no ritmo do aluno	[Llana et al. 2012]	1
G - Auto-aprendizagem	[Wang et al. 2011]	1
H - Motivação	[Georgouli and Guerreiro 2010]	1
I - Melhorar a estratégia pedagógica	[Rajala et al. 2016], [Vihavainen et al. 2013]	2
J - Oferecer uma visão global e temporal do desenvolvimento do aluno	[AlShamsi and Elnagar 2011]	1
K - Assiduidade do aluno	[Georgouli and Guerreiro 2010]	1
L - Transparência na correção	[Georgouli and Guerreiro 2010]	1

A Tabela 5 relaciona os sistemas de *Juiz Online* mencionados nos artigos com os benefícios identificados na Tabela 4. A descrição *\*Experiência UL-FCIS* foi incluída pelo fato do artigo não apresentar o nome do sistema, visto que o que aborda são as ideias de um protótipo produzido e os relatos da experiência.

**Tabela 5. Juizes Online relacionados aos benefícios**

SISTEMA	A	B	C	D	E	F	G	H	I	J	K	L
<i>Mooshak</i> [Georgouli and Guerreiro 2010]								X			X	X
<i>AutoLEP</i> [Wang et al. 2011]		X					X					
<i>eGrader</i> [AlShamsi and Elnagar 2011]		X								X		
<i>FLOP</i> [Llana et al. 2012]	X			X		X						
<i>Jutge.org</i> [Petit et al. 2012]	X		X									
<i>YOJ</i> [Sun et al. 2014]	X			X								
<i>Athene</i> [Pettit et al. 2015a]												
<i>*Experiência UL-FCIS</i> [Pozenel et al. 2015]												
<i>ViLLE</i> [Rajala et al. 2016]		X			X				X			
<i>ProgTest</i> [Souza et al. 2015]	X			X								
<i>Test My Code</i> [Vihavainen et al. 2013]	X								X			

**QP2: Quais os problemas ao usar Juiz Online no ensino de CS1?**

A Tabela 6 resume os principais problemas identificados nos trabalhos lidos. Grande parte deles estão relacionados à falta de validação, de maturidade dos sistemas e à dificuldade de instalação e manipulação dos programas. Existe também a preocupação com segurança e escalabilidade que impacta na gestão da infra-estrutura. Talvez devido à falta de maturidade e à origem baseada em competições, os sistemas apresentam problemas de usabilidade. O processo de submissão de programas ainda apresenta problemas, e o *feedback* não é suficiente para que alunos consigam corrigir muitos dos erros.

**Tabela 6. Problemas encontrados nos trabalhos**

PROBLEMA	REFERÊNCIAS	Nº TRABALHOS
A - Falta de maturidade dos sistemas	[AlShamsi and Elnagar 2011], [Pozenel et al. 2015]	2
B - Falta de validação do sistema	[Pettit et al. 2015b], [Llana et al. 2012], [Pettit et al. 2015a], [Pozenel et al. 2015]	4
C - Dificuldade em gerenciar a infra-estrutura	[Petit et al. 2012]	1
D - Usabilidade	[Souza et al. 2015]	1
E - Disponibilizar publicamente a nota de todos os alunos	[Georgouli and Guerreiro 2010]	1
F - Sistema contextualizado em competições	[Georgouli and Guerreiro 2010], [Sun et al. 2014]	2
G - Falta da informação sobre a contribuição dos sistemas para a aprendizagem	[Pettit et al. 2015b], [Georgouli and Guerreiro 2010], [Wang et al. 2011]	3
H - Sistema aumentou ocorrência de plágio	[Pozenel et al. 2015]	1
I - Aluno não testar seus programas	[Vihavainen et al. 2013]	1

Outros problemas surgem de decisões de implementação e das implicações do uso do sistema na sala de aula, e estão relacionados à interpretação dos usuários. Alguns alunos não gostam que as notas estejam visíveis para todos e nem do ambiente de competição que o uso de sistemas de Juiz Online criam. O fato do sistema fazer a correção automática pode levar a uma programação na base de tentativa e erro e ao aumento da ocorrência de plágio. Além do mais, não parece haver aumento das notas dos alunos [Wang et al. 2011].

A Tabela 7 relaciona os sistemas de Juiz Online com os problemas identificados na Tabela 6.

**QP3: Quais requisitos funcionais um Juiz Online precisa atender para ser usado no ensino de CS1?**

A leitura dos trabalhos sobre CS1 e Juiz Online trouxe quatro conjuntos de requisitos funcionais que podem variar conforme o contexto de ensino. A Tabela 8 mostra a

**Tabela 7. Juízes Online relacionados aos problemas**

SISTEMA	A	B	C	D	E	F	G	H	I
<i>Mooshak</i> [Georgouli and Guerreiro 2010]					X	X	X		
<i>AutoLEP</i> [Wang et al. 2011]							X		
<i>eGrader</i> [AlShamsi and Elnagar 2011]	X								
<i>FLOP</i> [Llana et al. 2012]		X							
<i>Jutge.org</i> [Petit et al. 2012]			X						
<i>YOJ</i> [Sun et al. 2014]						X			
<i>Athene</i> [Pettit et al. 2015a]		X							
<i>*Experiência UL-FCIS</i> [Pozenel et al. 2015]	X	X						X	
<i>ViLLE</i> [Rajala et al. 2016]									
<i>ProgTest</i> [Souza et al. 2015]				X					
<i>Test My Code</i> [Vihavainen et al. 2013]									X

distribuição de trabalhos quanto a esses conjuntos.

**Tabela 8. Requisitos funcionais encontrados nos trabalhos**

REQUISITO	REFERÊNCIAS	Nº TRABALHOS
C1 - <i>Feedback</i>	[Georgouli and Guerreiro 2010], [Wang et al. 2011], [AlShamsi and Elnagar 2011], [Llana et al. 2012] [Petit et al. 2012], [Sun et al. 2014], [Pettit et al. 2015a], [Pozenel et al. 2015] [Rajala et al. 2016], [Souza et al. 2015], [Vihavainen et al. 2013]	11
C2 - Integração do Sistema com os Cursos	[Wang et al. 2011], [Llana et al. 2012], [Petit et al. 2012], [Sun et al. 2014] [Pozenel et al. 2015], [Rajala et al. 2016], [Souza et al. 2015], [Vihavainen et al. 2013]	8
C3 - Análise do Desempenho Geral dos Alunos	[Petit et al. 2012], [AlShamsi and Elnagar 2011], [Vihavainen et al. 2013]	3
C4 - Oferecer Diferentes Tipos de Atividades	[Sun et al. 2014], [Pozenel et al. 2015], [Rajala et al. 2016], [Souza et al. 2015] [Vihavainen et al. 2013]	5

O *Feedback* é fundamental para a aprendizagem do aluno e requisito chave dos sistemas de *Juiz Online*. Normalmente o *Feedback* é o resultado de testes feitos pelo *Juiz Online* com o programa submetido pelo aluno [Wang et al. 2011, Llana et al. 2012, Petit et al. 2012, Sun et al. 2014, Pozenel et al. 2015, Rajala et al. 2016]. Entretanto, ele pode incluir pontuação parcial [Pettit et al. 2015a, Vihavainen et al. 2013], contribuição do professor [Georgouli and Guerreiro 2010, Sun et al. 2014, Vihavainen et al. 2013] ou de outros alunos [Sun et al. 2014]. Também pode ser escondido quando o sistema é usado em exames [Rajala et al. 2016].

Fica claro pelos artigos que existe um esforço em melhorar o tratamento de erros a partir do *feedback*, oferecendo uma melhor descrição dos erros de compilação e execução e da saída do programa [Rajala et al. 2016], oferecendo detalhes dos casos de teste que produziram erro [Pettit et al. 2015a], considerando aspectos de sintaxe [Wang et al. 2011] ou estruturais que se relacionam com a semântica [Wang et al. 2011, AlShamsi and Elnagar 2011], e identificando a evolução do estilo de codificação [Pettit et al. 2015a]. [Souza et al. 2015] propõe avaliar o programa do aluno com os casos de teste do professor e o programa escrito pelo professor com casos de teste do aluno, além de permitir ao professor a inclusão de dicas textuais.

Alguns trabalhos se preocuparam em integrar o sistema de *Juiz Online* com os cursos, o que pode significar novos requisitos. Existem estudos para integrá-los em um *IDE* para uso em *MOOCs* [Vihavainen et al. 2013], ou em sistemas de gestão de aprendizagem [Petit et al. 2012], flexibilizando a configuração,

o que facilita a adaptação do sistema às necessidades do curso [Sun et al. 2014, Souza et al. 2015], inclusive com a possibilidade de professores cadastrarem novos problemas [Llana et al. 2012, Souza et al. 2015]. Além disso, os sistemas podem ser usados em exames [Wang et al. 2011, Petit et al. 2012, Rajala et al. 2016], e permitem acoplar a detecção de plágio [Sun et al. 2014, Pozenel et al. 2015]. A organização de problemas por nível de dificuldade foi considerada em [Petit et al. 2012, Rajala et al. 2016].

Problemas que exigem dos alunos sua resolução por meio da escrita de programas são os mais comuns nos Juizes *Online*. Porém, outros tipos de atividades foram identificadas nos trabalhos, como por exemplo problemas que possuem parte do código fonte escrito e solicitam ao aluno que resolva o restante [Sun et al. 2014, Rajala et al. 2016], o envio de casos de teste [Rajala et al. 2016, Vihavainen et al. 2013] e a criação de classes com métodos que seguem assinaturas pré-definidas [Pozenel et al. 2015]. [Pozenel et al. 2015] apresentaram problemas de CS1 envolvendo computação gráfica. Já [Rajala et al. 2016] propõem problemas dos tipos (*Quiz*, *Code shuffling*, Exercício Robô e Ligue os Itens). Existem sistemas que emitem relatórios com o desempenho de alunos [AlShamsi and Elnagar 2011, Petit et al. 2012, Vihavainen et al. 2013] ou turmas [AlShamsi and Elnagar 2011].

A Tabela 9 relaciona os sistemas de Juiz *Online* com os requisitos funcionais identificados que foram apresentados na Tabela 8.

**Tabela 9. Juizes *Online* relacionados aos requisitos funcionais**

SISTEMA	Feedback	Integração com os Cursos	Desempenho de Alunos	Diferentes Atividades
<i>Mooshak</i> [Georgouli and Guerreiro 2010]	X			
<i>AutoLEP</i> [Wang et al. 2011]	X	X		
<i>eGrader</i> [AlShamsi and Elnagar 2011]	X		X	
<i>FLOP</i> [Llana et al. 2012]	X	X		
<i>Jutge.org</i> [Petit et al. 2012]	X	X	X	
<i>YOJ</i> [Sun et al. 2014]	X	X		X
<i>Athene</i> [Pettit et al. 2015a]	X			
<i>*Experiencia UL-FCIS</i> [Pozenel et al. 2015]	X	X		X
<i>ViLLE</i> [Rajala et al. 2016]	X	X		X
<i>ProgTest</i> [Souza et al. 2015]	X	X		X
<i>Test My Code</i> [Vihavainen et al. 2013]	X	X	X	X

#### QP4: Quais requisitos não-funcionais um Juiz *Online* precisa atender para ser usado no ensino de CS1?

Os requisitos não-funcionais que o sistema precisa atender são dependentes dos requisitos funcionais definidos. A Tabela 10 mostra a distribuição dos requisitos não-funcionais encontrados nos trabalhos.

**Tabela 10. Requisitos não-funcionais encontrados nos trabalhos**

REQUISITO	REFERÊNCIAS	Nº TRABALHOS
Integração	[Vihavainen et al. 2013, Georgouli and Guerreiro 2010]	2
Usabilidade	[Llana et al. 2012]	1
Segurança	[Llana et al. 2012, Petit et al. 2012, Pozenel et al. 2015, Rajala et al. 2016]	4
Escalabilidade	[Petit et al. 2012, Pozenel et al. 2015]	2
Disponibilidade	[Vihavainen et al. 2013, Llana et al. 2012, Sun et al. 2014]	3

Existe um grande interesse em desenvolver sistemas de Juiz *Online* para serem amplamente usados na *internet* [Llana et al. 2012, Petit et al. 2012, Pozenel et al. 2015,

Rajala et al. 2016]. Nesse contexto, alguns aspectos, principalmente de segurança, devem ser considerados, como o tratamento de programas mal-intencionados [Llana et al. 2012, Petit et al. 2012] e o uso de *firewall* [Rajala et al. 2016]. [Pozenel et al. 2015] propõem a execução dos programas submetidos pelos alunos como um usuário sem privilégios no *linux* para lidar com a segurança.

A escalabilidade foi tratada pelos trabalhos [Petit et al. 2012, Pozenel et al. 2015], e deve ser considerada quando se tem o interesse em um sistema com quantidade crescente de usuários e transações, implicando no requisito não-funcional disponibilidade [Vihavainen et al. 2013, Llana et al. 2012, Sun et al. 2014], especialmente presente quando usados em *MOOCs*.

A Tabela 11 relaciona os sistemas de Juiz *Online* com os requisitos não-funcionais identificados. Os requisitos não-funcionais descritos são integração, usabilidade, segurança, escalabilidade, e disponibilidade. Os dados mostram que não há uma forte preocupação com esses requisitos, o que pode ser devido aos sistemas ainda não terem uma certa maturidade. Acredita-se que quando houver uma popularização maior dos sistemas no ensino, os projetos buscarão atacar mais tais questões.

**Tabela 11. Juízes *Online* relacionados aos requisitos não-funcionais**

SISTEMA	Integ.	Usabil.	Seg.	Escalab.	Disponib.
<i>Mooshak</i> [Georgouli and Guerreiro 2010]	X				
<i>AutoLEP</i> [Wang et al. 2011]					
<i>eGrader</i> [AlShamsi and Elnagar 2011]					
<i>FLOP</i> [Llana et al. 2012]		X	X		X
<i>Jutge.org</i> [Petit et al. 2012]			X	X	
<i>YOJ</i> [Sun et al. 2014]					X
<i>Athene</i> [Pettit et al. 2015a]					
<i>*Experiência UL-FCIS</i> [Pozenel et al. 2015]			X	X	
<i>ViLE</i> [Rajala et al. 2016]			X		
<i>ProgTest</i> [Souza et al. 2015]					
<i>Test My Code</i> [Vihavainen et al. 2013]	X				X

#### 4. Conclusão

Para responder à questão *Qual é a especificação mais adequada para que um juiz online atenda à disciplina de CSI?* é necessário explorar diferentes configurações de requisitos de sistemas de Juiz *Online* nos diferentes contextos educacionais e levantar a contribuição para a aprendizagem do aluno. A variedade de requisitos funcionais e não-funcionais possíveis é grande, como apresenta esta RSL. A leitura dos artigos selecionados possibilitou classificar os requisitos funcionais em quatro grupos (*feedback*, integração do sistema com os cursos, análise do desempenho geral dos alunos, e oferecer diferentes tipos de atividades) e os requisitos não-funcionais (integração, usabilidade, segurança, escalabilidade, disponibilidade).

A questão da eficácia educacional das ferramentas de Juiz *Online* ainda é um problema em aberto. [Ramos et al. 2015] mostraram que somente 10.2% dos artigos avaliados em seu trabalho apresentaram rigor nos resultados, trazendo a validação das ferramentas como um desafio. No entanto, esta RSL apontou diversos benefícios e problemas que podem ocorrer no uso dessas ferramentas. Essa análise é importante para saber quais são os problemas que precisam ser resolvidos e apoiar na busca dos benefícios. Aprendizagem no ritmo do aluno, auto-aprendizagem e redução da carga de trabalho do professor,

são alguns dos benefícios apontados que contribuem não só em ambientes tradicionais de ensino, mas em ambientes de Educação a Distância (EAD) e em MOOC's. A liberdade de definir listas de exercício e a disponibilidade de instrumentos para acompanhar os alunos são questões importantes para o professor.

O *Feedback* é fundamental nos sistemas de Juiz *Online*. Existem projetos que visam melhorá-lo, trazendo informações sobre aspectos de sintaxe ou estruturais que se relacionam com a semântica e possibilitando identificar a evolução do estilo de codificação do aluno. Isso traz novas perspectivas para a pesquisa.

O levantamento feito aponta a importância de analisar a qualidade de ferramentas educacionais. É necessário definir padrões que possibilitem comparar as ferramentas nos diferentes contextos de ensino em uma abordagem mais formal e rigorosa. Vale observar também, que existem sistemas de Juiz *Online* que estão sendo produzidos e usados no Brasil com bastante popularidade, como o *BOCA*<sup>1</sup> e o *URI*<sup>2</sup>, que não foram tratados nesta RSL por não terem sido retornados na pesquisa realizada conforme protocolo definido.

## Referências

- AlShamsi, F. and Elnagar, A. (2011). An automated assessment and reporting tool for introductory java programs. In *Innovations in Information Technology (IIT), 2011 International Conference on*, pages 324–329. IEEE.
- Byrne, P. and Lyons, G. (2001). The effect of student attributes on success in programming. In *ACM SIGCSE Bulletin*, volume 33, pages 49–52. ACM.
- Georgouli, K. and Guerreiro, P. (2010). Incorporating an automatic judge into blended learning programming activities. In *Advances in Web-Based Learning–ICWL 2010*, pages 81–90. Springer.
- Gomes, A., Areias, C., Henriques, J., and Mendes, A. J. (2008). Aprendizagem de programação de computadores: dificuldades e ferramentas de suporte. *Revista Portuguesa de Pedagogia*, 42(2).
- Kitchenham, B. (2004). Procedures for performing systematic reviews. *Keele, UK, Keele University*, 33(2004):1–26.
- Kurnia, A., Lim, A., and Cheang, B. (2001). Online judge. *Computers & Education*, 36(4):299 – 315.
- Llana, L., Martin-Martin, E., and Pareja-Flores, C. (2012). Flop, a free laboratory of programming. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*, pages 93–99. ACM.
- Petit, J., Giménez, O., and Roura, S. (2012). Judge.org: an educational programming judge. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*, pages 445–450. ACM.
- Pettit, R., Homer, J., Gee, R., Mengel, S., and Starbuck, A. (2015a). An empirical study of iterative improvement in programming assignments. In *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, pages 410–415. ACM.

<sup>1</sup>**BOCA**: disponível em [www.ime.usp.br/~cassio/boca/](http://www.ime.usp.br/~cassio/boca/)

<sup>2</sup>**URI**: disponível em [www.urionlinejudge.com.br](http://www.urionlinejudge.com.br)

- Pettit, R., Homer, J., Holcomb, K., Simone, N., and Mengel, S. (2015b). Are automated assessment tools helpful in programming courses? *122nd ASEE Annual Conference and Exposition. American Society for Engineering Education, 2015*.
- Pozenel, M., Furst, L., and Mahnicc, V. (2015). Introduction of the automated assessment of homework assignments in a university-level programming course. In *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*, pages 761–766. IEEE.
- Rajala, T., Kaila, E., Lindén, R., Kurvinen, E., Lökkila, E., Laakso, M.-J., and Salakoski, T. (2016). Automatically assessed electronic exams in programming courses. In *Proceedings of the Australasian Computer Science Week Multiconference*, page 11. ACM.
- Ramos, V., Wazlawick, R., Galimberti, M., Freitas, M., and Mariani, A. C. (2015). A comparação da realidade mundial do ensino de programação para iniciantes com a realidade nacional: Revisão sistemática da literatura em eventos brasileiros. In *Anais do Simpósio Brasileiro de Informática na Educação*, volume 26, page 318.
- Rountree, N., Rountree, J., Robins, A., and Hannah, R. (2004). Interacting factors that predict success and failure in a cs1 course. *ACM SIGCSE Bulletin*, 36(4):101–104.
- Souza, D. M. D., Isotani, S., and Barbosa, E. F. (2015). Teaching novice programmers using progtest. *International Journal of Knowledge and Learning*, 10(1):60–77.
- Sun, H., Li, B., and Jiao, M. (2014). Yoj: An online judge system designed for programming courses. In *Computer Science & Education (ICCSE), 2014 9th International Conference on*, pages 812–816. IEEE.
- Vieira, M. A. (2016). *Modelagem de Espaços Inteligentes Pessoais e Espaços Inteligentes Fixos no contexto de Cenários de Computação Ubíqua - Dissertação de Mestrado em Ciência da Computação*. PhD thesis, Universidade Federal de Goiás.
- Vier, J., Gluz, J., and Jaques, P. A. (2015). Empregando redes bayesianas para modelar automaticamente o conhecimento dos alunos em lógica de programação. *Revista Brasileira de Informática na Educação*, 23(02):45.
- Vihavainen, A., Luukkainen, M., and Pärtel, M. (2013). Test my code: An automatic assessment service for the extreme apprenticeship method. In *2nd International Workshop on Evidence-based Technology Enhanced Learning*, pages 109–116. Springer.
- Wang, T., Su, X., Ma, P., Wang, Y., and Wang, K. (2011). Ability-training-oriented automated assessment in introductory programming course. *Computers & Education*, 56(1):220–226.
- Wilson, B. C. and Shrock, S. (2001). Contributing to success in an introductory computer science course: a study of twelve factors. In *ACM SIGCSE Bulletin*, volume 33, pages 184–188. ACM.